



**UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**

INGENIERÍA TÉCNICA DE INFORMÁTICA DE GESTIÓN

PROYECTO FIN DE CARRERA



**ENTORNO Y APLICACIÓN MÓVIL ANDROID
“HISPANIA UC3M”**

AUTOR: Rafael González Grönberg

TUTOR: Miguel Ángel Patricio

30 de Julio de 2015

Agradecimientos

Ante este proyecto, me gustaría agradecer la paciencia que ha tenido mi madre aguantándome cada día durante el periodo que le he dedicado a la realización de este, su ayuda y apoyo incondicional me han allanado enormemente el camino permitiéndome centrarme en la realización de este.

A mi novia Ana ayudándome a manejar los programas de edición de imagen que tantos quebraderos de cabeza me generan.

Me gustaría hacer especial mención a mi tutor don Miguel Ángel Patricio, pues Jamás he recibido otra cosa que palabras de apoyo y ayuda para la consecución de los objetivos, no sólo en la realización de este proyecto si no desde que tuve la suerte de ser su alumno y poder aprender un poquito cada día de sus enseñanzas.

Finalmente me gustaría dedicarle mi esfuerzo a mi Padre. El año pasado perdió la batalla contra el Cáncer y sé que haya donde esté se sentirá orgulloso de ver como su hijo, por fin, termina la carrera. Gracias papá por enseñarme todo lo que sé y darme, aunque duras, tan valiosas lecciones que me hacen apreciar la vida desde una perspectiva mucho más madura.

Son las pequeñas cosas las que hacen las grandes cosas posibles. Solo una cuidada atención sobre los pequeños detalles de cualquier proyecto hacen que se convierta en un proyecto de primera clase

J. Willard Marriott
1900-1985 empresario Estadounidense

El hombre inteligente es capaz de crear más oportunidades que las que salen a su encuentro

Francis Bacon
1561-1626 filósofo y escritor Inglés

La técnica es el esfuerzo para ahorrar esfuerzo.

José Ortega y Gasset
(1883-1955) Filósofo y ensayista español.

RESUMEN

La aparición de nuevas tecnologías y las mayores capacidades de cómputo en un espacio cada vez más reducido nos sugieren cada vez un mundo más interconectado. Un ejemplo de esta tendencia es la aparición y masificación de los denominados “smartPhones” sobre los cuales, y su plataforma más extendida, Android, basaremos la realización de este proyecto.

En el caso de este trabajo trato de generar una herramienta que sea la base para un sistema de aprendizaje y ocio basado en la gamificación. He optado por utilizar un repertorio de preguntas basado en Historia de España, por considerarlo un tema de interés cultural general, no obstante con pequeñas modificaciones se podría adaptar a cualquier temática.

INDICE

1. Tabla de contenido

1. INTRODUCCIÓN	11
1.1. Objetivos	12
1.2. Estructura del proyecto	14
2. ESTADO DEL ARTE	16
2.1. El Smartphone, historia y tecnología.....	16
2.2. Hardware	27
2.2.1. CPU & GPU	27
2.2.2. hardware complementario	29
2.2.3. Sensores.....	30
2.2.4. Conectividad a internet	31
2.3. Sistemas Operativos.....	32
2.3.1. Android	32
2.3.2. iOS	38
2.3.3. Windows Phone & Windows 10	43
2.3.4. Otros.....	44
2.3.5. Comparativa de mercado de los principales Sistemas operativos	46
2.4. Los diferentes mercados de Aplicaciones	47
2.5. Consideraciones específicas para HISPANIA UC3M.....	50
2.5.1. Entorno Android	50
2.5.2. Hardware necesario.....	51
2.5.3. Google Play	51
2.6. Diseño de la Aplicación Móvil.....	52
2.6.1. Elementos Clave de la interfaz basada en usuario.....	52
2.6.2. Usabilidad de la aplicación.....	53
2.7. Aplicaciones Móviles: Juegos, educación y entretenimiento.	54
2.7.1. Logos Quiz.....	54
2.7.2. QuizUp	56
2.7.3. Pasopalabra.....	57
2.7.4. El Juego de las Adivinanzas.....	57
3. ANDROID	58
3.1. Arquitectura	58
3.1.1. El núcleo Linux.....	59
3.1.2. Android Runtime.....	59
3.1.3. Librerías Nativas.....	61
3.1.4. Framework de Aplicaciones.....	61

3.1.5. Aplicaciones	62
3.2. El patrón MVC.....	63
3.3. Elementos que componen una aplicación Android.	64
3.3.1. Activity	64
3.3.2. Servicios.....	65
3.3.3. Content Providers.....	65
3.3.4. Intents.....	66
3.3.5. Broadcast Receivers.....	66
3.3.6. Widgets	67
3.3.7. Notificaciones	67
3.3.8. El archivo Manifest.xml	67
3.3.9. Recursos estáticos y localización	68
3.3.10. Otros Recursos y localizaciones.....	70
3.4. Fragmentos	71
3.5. Threads y Async-Task	73
3.6. El Cliente HTTP	74
4. HERRAMIENTAS DE DESARROLLO	75
4.1. Herramientas de diseño	75
4.1.1. Lápiz y papel.....	75
4.1.2. Adobe Photoshop.....	76
4.2. Herramientas de Programación.....	78
4.2.1. Eclipse.....	79
4.2.2. Herramientas de tratamiento de Datos.....	81
5. HISPANIA UC3M, ANALISIS Y DISEÑO E IMPLEMENTACIÓN	82
5.1. ANÁLISIS.....	82
5.1.1. Identificación del Entorno tecnológico.....	82
5.1.2. Ciclo de Vida.....	84
5.1.3. Requisitos Software.....	85
5.1.4. Requisitos de Restricción.....	93
5.1.5. Casos de Uso.....	95
5.1.6. Diagramas de secuencia	99
5.2. DISEÑO	106
5.2.1. Modelo de datos	106
5.2.2. Esquema Aplicación; Navegación de pantallas y activitys.....	109
5.3. IMPLEMENTACIÓN.....	111
5.3.1. Sentando las bases del desarrollo incremental- GIT.....	111
5.3.2. Sistemas Backend.....	112
5.3.3. Tecnologías y bibliotecas utilizadas en la App.	121
6. GAMIFICACIÓN, REDES SOCIALES Y SHARING	135
6.1. Integración con Facebook.....	135

6.1.1. La consola de FB.....	136
6.1.2. Integrando el código de FB.	139
6.2. Integración con Twitter.....	140
6.3. Integración con e-mail, SMS y WhatsApp.	141
7. CONTROL DE CALIDAD.....	142
7.1. Evaluación usabilidad	142
7.1.1. Perfiles evaluador.....	143
7.1.2. Pruebas.....	143
7.1.3. resultados.....	144
7.2. Modificaciones de mejora	145
8. PLANIFICACIÓN Y PRESUPUESTO	147
8.1. Planificación	147
8.1.1. Identificación de Tareas.....	147
8.1.2. Diagrama de Gantt.....	149
8.1.3. Diagrama de Recursos	152
8.2. Presupuesto	153
9. LÍNEAS DE DESARROLLO FUTURAS.....	157
9.1. Apertura de nuevos mercados	157
9.2. Mejoras sobre la infraestructura existente.....	158
9.3. back-end.....	158
10. Bibliografía y referencias.....	159
11. APENDICE I – LICENCIA CREATIVE COMMONS	164
12. APENDICE II – LICENCIA GPL	173
13. <u>APENDICE III - CUENTAS Y DATOS DE ACCESO</u>	180
14. <u>APENDICE IV – ICONOGRAFÍA E IMAGOTIPO:</u>.....	181

Referencias de Ilustraciones

<i>Ilustración 1: Historia , IBM Simón.....</i>	<i>17</i>
<i>Ilustración 2 - Historia , Ericsson GS88.....</i>	<i>17</i>
<i>Ilustración 3 Historia, Ericsson R380</i>	<i>18</i>
<i>Ilustración 4, Historia, BlackBerry 850.....</i>	<i>18</i>
<i>Ilustración 5, Historia, Iphone 1.....</i>	<i>19</i>
<i>Ilustración 6- Historia, HTC Dream.....</i>	<i>20</i>
<i>Ilustración 7- Historia Motorola Droid.....</i>	<i>20</i>
<i>Ilustración 8 Historia, Tabla Dispositivos Apple</i>	<i>21</i>
<i>Ilustración 9 – Historia, Tabla Dispositivos Samsung.....</i>	<i>22</i>
<i>Ilustración 10 – Historia, Infografía Samsung.....</i>	<i>23</i>
<i>Ilustración 11 - Historia, Infografía SmartPhones 1.....</i>	<i>24</i>
<i>Ilustración 12 - Historia, Infografía SmartPhones 2.....</i>	<i>25</i>
<i>Ilustración 14 - Historia, Infografía SmartPhones 3.....</i>	<i>26</i>
<i>Ilustración 15, E.A-Tabla rendimiento por Core.....</i>	<i>27</i>
<i>Ilustración 16 E.A- Porcentaje Mercado por Fabricante chips.....</i>	<i>28</i>
<i>Ilustración 17 E.A- Detalle funcionamiento A-GPS.....</i>	<i>29</i>
<i>Ilustración 18 E.A- Redes de comunicación.....</i>	<i>31</i>
<i>Ilustración 19 E.A- Detalle arquitectura Capas Android</i>	<i>32</i>
<i>Ilustración 20 E.A- Detalle Versiones Android</i>	<i>33</i>
<i>Ilustración 21 E.A- Tabla detallada Versiones Android.....</i>	<i>35</i>
<i>Ilustración 22 E.A- Fragmentación Temporal Dispositivos Android.....</i>	<i>36</i>
<i>Ilustración 23 E.A- Detalle Fragmentación Android Mayo 2015.....</i>	<i>37</i>
<i>Ilustración 24 E.A- Detalle Capas S.O. Ios.....</i>	<i>38</i>
<i>Ilustración 25 E.A- Tabla Versiones Ios.....</i>	<i>41</i>
<i>Ilustración 26 E.A- Línea Histórica comparativa versiones Android / Ios.....</i>	<i>42</i>
<i>Ilustración 27 E.A- tabla comparativa Versiones Windows Phone.....</i>	<i>43</i>
<i>Ilustración 28 E.A- Tabla Cuota Mercado S.O.....</i>	<i>46</i>
<i>Ilustración 29 E.A- Tabla resumen Diferencias S.O.....</i>	<i>46</i>
<i>Ilustración 30 E.A-Gráfico Aplicaciones Disponibles Google Play</i>	<i>47</i>
<i>Ilustración 31 E.A- Gráfico aplicaciones Disponible AppStore</i>	<i>48</i>
<i>Ilustración 32 E.A- Detalle Aplicaciones Amazon AppStore</i>	<i>49</i>
<i>Ilustración 33 E.A- Logo, LogosQuiz.....</i>	<i>54</i>
<i>Ilustración 34 E.A- Detalles LogosQuiz</i>	<i>55</i>
<i>Ilustración 35 E.A- Logo QuizUp.....</i>	<i>56</i>
<i>Ilustración 36 y 37 Detalle QuizUp</i>	<i>56</i>
<i>Ilustración 38, 39 y40 Detalle Pasopalabra.....</i>	<i>57</i>
<i>Ilustración 41 Detalle Adivinanzas.....</i>	<i>57</i>
<i>Ilustración 42 Diseño Capas Android.....</i>	<i>58</i>
<i>Ilustración 43 Detalle Fichero DEX.....</i>	<i>59</i>
<i>Ilustración 44Tabla Comparativa DALVIK- ART.....</i>	<i>60</i>
<i>Ilustración 45 Comparativa rendimiento Dalvik- ART.....</i>	<i>60</i>
<i>Ilustración 46 Tabla detalle Modelo MVC.....</i>	<i>63</i>

<i>Ilustración 47 Detalle MVC</i>	<i>63</i>
<i>Ilustración 48 Detalle Activity</i>	<i>64</i>
<i>Ilustración 49 Ciclo Vida Servicio</i>	<i>65</i>
<i>Ilustración 50 y 51 Diseño Content Providers</i>	<i>65</i>
<i>Ilustración 52 BroadCast Receiver</i>	<i>66</i>
<i>Ilustración 53 ejemplo widget</i>	<i>67</i>
<i>Ilustración 54 Notificacion Reducida</i>	<i>67</i>
<i>Ilustración 55 Notificación Completa</i>	<i>67</i>
<i>Ilustración 56 Detalle Fragmentos</i>	<i>71</i>
<i>Ilustración 57 Ciclo de Vida de Fragmento</i>	<i>72</i>
<i>Ilustración 58 Detalle Thread</i>	<i>73</i>
<i>Ilustración 59 Detalle Asynk Task</i>	<i>73</i>
<i>Ilustración 60 Tabla Ilustraciones y Cambios</i>	<i>77</i>
<i>Ilustración 61 Ilustración filtrado de Ides</i>	<i>78</i>
<i>Ilustración 62 Tabla recursos Hardware utilizados</i>	<i>83</i>
<i>Ilustración 63 Detalle Ciclo de Vida</i>	<i>84</i>
<i>Ilustración 64 Tablas Requisitos Funcionales</i>	<i>92</i>
<i>Ilustración 65 Tablas Requisitos Restricción</i>	<i>94</i>
<i>Ilustración 66 Diagrama de Casos de Uso</i>	<i>95</i>
<i>Ilustración 67 Tablas de Casos de Uso</i>	<i>98</i>
<i>Ilustración 68 Caso de Uso Servidor</i>	<i>98</i>
<i>Ilustración 69 Tabla Casos Uso Servidor</i>	<i>98</i>
<i>Ilustración 70 Diagrama de secuencia CU1</i>	<i>99</i>
<i>Ilustración 71 Diagrama de secuencia CU 2</i>	<i>100</i>
<i>Ilustración 72 Diagrama de secuencia CU 3</i>	<i>100</i>
<i>Ilustración 73 Diagrama de secuencia CU 4</i>	<i>101</i>
<i>Ilustración 74 Diagrama de secuencia CU 5</i>	<i>102</i>
<i>Ilustración 75 Diagrama de secuencia CU6</i>	<i>102</i>
<i>Ilustración 76 Diagrama de secuencia CU 7</i>	<i>103</i>
<i>Ilustración 77 Diagrama de secuencia CU 8</i>	<i>104</i>
<i>Ilustración 78 Diagrama de secuencia CU 9</i>	<i>105</i>
<i>Ilustración 79 Diagrama de secuencia CU 10</i>	<i>105</i>
<i>Ilustración 80 Modelo de Datos SQLite Local</i>	<i>106</i>
<i>Ilustración 81 Modelo Datos Usuario</i>	<i>107</i>
<i>Ilustración 82 Modelo de Datos Backend 42, Lógica de Juego</i>	<i>108</i>
<i>Ilustración 83 Esquema de Pantallas y actividades</i>	<i>110</i>
<i>Ilustración 84 Detalla Vista Source Tree</i>	<i>111</i>
<i>Ilustración 85 Tabla Características Game Sparks</i>	<i>113</i>
<i>Ilustración 86 Tabla Características Brain Cloud</i>	<i>114</i>
<i>Ilustración 87 Tabla Características App 42</i>	<i>115</i>
<i>Ilustración 88 Tabla Características Parse</i>	<i>119</i>
<i>Ilustración 89 Detalle Dialogo Espera</i>	<i>126</i>
<i>Ilustración 90 Detalle bloqueo interfaz multijugador</i>	<i>126</i>
<i>Ilustración 91 Detalle Activación GCM I</i>	<i>128</i>
<i>Ilustración 92 Detalle Activación GCM II</i>	<i>128</i>
<i>Ilustración 93 y 94 Detalle Activación GCM en APP42</i>	<i>128</i>

<i>Ilustración 95 Captura Pantalla Google Analytics</i>	131
<i>Ilustración 96 Captura Pantalla Google Analytics- Audiencia</i>	132
<i>Ilustración 97 Captura Pantalla Google Analytics- Comportamiento</i>	132
<i>Ilustración 98 Captura Pantalla Google AdMob</i>	134
<i>Ilustración 99 Captura Pantalla ingresos AdMob</i>	134
<i>Ilustración 100 Tabla ilustrada de pantallas y funcionalidades Facebook</i>	138
<i>Ilustración 101 Captura pantalla configuración Twitter</i>	140
<i>Ilustración 102 Tabla Perfiles Evaluador</i>	143
<i>Ilustración 103 Tabla Pruebas a Realizar</i>	143
<i>Ilustración 104 Tabla Resultados Control de calidad</i>	144
<i>Ilustración 105 Modificación propuesta en control de calidad I</i>	145
<i>Ilustración 106 Modificación propuesta en control de calidad II</i>	145
<i>Ilustración 107 Modificación propuesta en control de calidad III</i>	146
<i>Ilustración 108 Tabla Tareas</i>	148
<i>Ilustración 109 : Detalle Gantt Global</i>	149
<i>Ilustración 110 : Gantt Fase 0</i>	149
<i>Ilustración 111 : Gantt Fase 1</i>	150
<i>Ilustración 112 : Gantt Fase 2</i>	150
<i>Ilustración 113 : Gantt Fase 3</i>	151
<i>Ilustración 114 : Gantt Control de calidad y despliegue</i>	151
<i>Ilustración 115 Diagrama de recursos</i>	152
<i>Ilustración 116 tabla Costes / Recurso</i>	153
<i>Ilustración 117 Recursos en Horas Fase 0</i>	153
<i>Ilustración 118 Recursos en Horas Fase 1</i>	153
<i>Ilustración 119 Recursos en Horas Fase 2</i>	154
<i>Ilustración 120 Recursos en Horas Fase 3</i>	154
<i>Ilustración 121 Recursos en Horas Control Calidad</i>	154
<i>Ilustración 122 Recursos en Horas Despliegue</i>	154
<i>Ilustración 123 Tabla coste por recurso</i>	155
<i>Ilustración 124 Tabla Costes totales</i>	155
<i>Ilustración 125 Tabla Coste Mantenimiento</i>	156
<i>Ilustración 126 Logo Hispania Azul uc3m</i>	181
<i>Ilustración 127 Logo Hispania B/N</i>	181
<i>Ilustración 128 - Logo Hispania Rectangular</i>	181
<i>Ilustración 129 Logo Hispania amarillo V.1</i>	181
<i>Ilustración 130 Logo Hispania amarillo V.2</i>	181
<i>Ilustración 131 Imagen Destacada para Google Play</i>	181

CAPÍTULO 1

INTRODUCCIÓN

1. INTRODUCCIÓN

Actualmente , en pleno auge de la sociedad de la comunicación donde cada vez son más los sistemas y dispositivos que funcionan de manera conectada se nos plantean la necesidad de utilizar estos medios en pro de una mejor sociedad del conocimiento. Esta situación nos justifica el interés de realizar una base sólida sobre este entorno incorporando perspectivas epistemológicas y metodológicas con el fin de desempeñar el diseño adaptado hacia la nueva sociedad que se perfila.

El proyecto “Hispania UC3M” tiene como objetivo fundamental desarrollar una base operativa para el despliegue posterior sobre el entorno móvil de plataformas de ámbito o bien educacional o enfocadas al ocio en donde la base común sea contribuir a la adquisición de conocimientos mediante una estructura gamificada.

Las principales fases a seguir han sido:

- 1.- Estudio de las diferentes plataformas sobre las que se puede desarrollar el proyecto, pros y contras de cada una.
- 2.- Diseño del sistema de Gamificación, en donde la simplicidad de cara al usuario final nos permita llegar a el mayor rango de público posible, requisito deseable debido a la naturaleza pedagógico-social de la herramienta.
- 3.- Estudio y estructuración de los datos , elección del backend.
- 4.- Diseño de interfaz de usuario.
- 5.- Programación.
- 6.- Creación e integración de cuentas en redes Sociales y que aporten valor añadido a la gamificación.

Otras consideraciones que se han tenido en cuenta durante todo el desarrollo es la interoperabilidad, tanto hipotética como futurible con otros sistemas sobre los que se pueda llevar un nuevo despliegue de la plataforma.

Como breve marco-resumen del proyecto descrito en el presente documento, estamos ante el desarrollo de una herramienta de transmisión de conocimientos , ya sea con un fin pedagógico o de entretenimiento educativo y tal como se describirá posteriormente para este fin nos apoyaremos en las nuevas plataformas Mobile aprovechando sus recursos y creando un sistema escalable, de rápida implementación y con una interfaz enfocada a la facilidad de uso.

1.1. Objetivos

Partiendo de las necesidades identificadas dentro del marco del proyecto HISPANIA UC3M , los principales requisitos a satisfacer por el sistema propuesto pueden resumirse en:

- **Desarrollo de una Aplicación Móvil** en base al sistema gamificado obtenido en la fase de concepción del proyecto. Esta aplicación debe ofrecer , entre otras, las siguientes funcionalidades.
 - ✓ **Autenticación de Usuarios y Control de acceso:** sólo se permitirá el acceso a el sistema a los miembros que previamente hayan sido registrados y cada usuario sólo podrá ver los rankings generales, de otros usuarios que previamente hayan sido marcado como amigos y los propios.
 - ✓ **Interfaz sencillo orientado a la experiencia de usuario:** el diseño es claro y sencillo, permitiendo a los usuarios menos experimentados utilizar nuestro software sin tener que pasar una etapa de aprendizaje. Esta es una condición *sine qua non*, pues una elevada dificultad de manejo aumentaría la tasa de rechazo y mermaría el efecto viralizante que una aplicación gamificada busca.
- **Selección y creación del Backend.** Consiste en el diseño e implementación de la infraestructura sobre la que funcionará nuestra app.
 - ✓ **Selección del backend del sistema Gamificado:**
Se seleccionará el más acorde a las necesidades del proyecto para que el sistema de juego, puntuaciones y datos de usuario sea funcional y plenamente compatible con otras plataformas que a posteriori se pudiesen agregar.
 - ✓ **Creación del Modelo de datos de comunicación cliente-Servidor:**
Establecemos un meta data para que las comunicaciones puedan ser independientes de plataforma, y que cuenten con todos los campos necesarios para la gestión de los datos necesarios de nuestro software.

✓ **Establecer servicios o servidores auxiliares:**

Se generarán servicios auxiliares que ayuden a maximizar las capacidades tanto operativas como funcionales de nuestra aplicación, apoyándonos en servidores auxiliares, en donde la finalidad es la optimización de los recursos y aprovechar las mejores capacidades de los diferentes servicios utilizados.

Este proyecto está realizado en todo momento sobre la idea de conocimiento divulgativo, pues ha sido desarrollado íntegramente con herramientas de software libre. Desde el punto de vista de la aplicación de licencias todos los artefactos están sujetos a la aplicación de la licencia GPL (Free Software-Fundation, 1991) y están recogidos en el presente documento en su apéndice primero. No obstante, la presente memoria estará supeditada a la licencia Creative Commons Reconocimiento-CompartirIgual 2.5 España disponible en Creative Commons (2007), la cual podemos encontrar en el apéndice segundo.

1.2. Estructura del proyecto

El proyecto se ha organizado de fuera hacia dentro, es decir, desde un estudio del entorno hasta llegar finalmente a un presupuesto de la aplicación creada.

En este capítulo, **capítulo 1**, se ha ofrecido una introducción a el proyecto, sus objetivos y un breve resumen del trabajo a desarrollar, se han presentado las motivaciones y unas sencillas guías fundamentales sobre las que se debe regir la totalidad del proyecto.

En el nuevo capítulo que procedemos a abordar, **capítulo 2**, estudiaremos el estado del arte en las materias con las que se relaciona este proyecto; conceptos sobre las aplicaciones móviles, enfocadas principalmente al entorno Android, seleccionado por ser más abierto y llegar a un mayor número de usuarios, conceptos sobre backend y los diferentes servicios auxiliares que se implementan así como del desarrollo software en su conjunto.

Posteriormente, **capítulo 3**, se realizará un estudio detallado del Sistema Android, esto genera la base teórica sobre la que se sustenten los conocimientos necesarios para la realización práctica del proyecto.

En el **capítulo 4**, describiremos las herramientas utilizadas en la elaboración del proyecto. Empezando desde las más básicas como sencillos formularios o mock-ups que nos han ayudado a decidirnos hacia el sistema de gamificación, diseño y pantallas, a las específicas utilizadas para el desarrollo integro de la aplicación móvil en la plataforma Android.

A continuación, en base a los conocimientos teóricos adquiridos y las herramientas necesarias, procedemos a elaboración del desarrollo software propio del aplicativo. En el **capítulo 5** se expondrá el análisis, diseño e implementación, es decir se detallara la fase productiva tanto de la app física como de los entornos necesarios para su correcta ejecución.

Seguido del desarrollo puro, se pasará en el **capítulo 6** a detallar como ha sido la implementación específica de las diferentes redes sociales y modelos de sharing sobre las que se apoya el aplicativo.

Una vez se posee un producto desarrollado, no es coherente lanzarlo al mercado sin realizar un control de calidad, el cual se detallará en el **capítulo 7**, se realizará un estudio con usuarios reales y sobre él se estudiarán mejoras que ayuden a mejorar la aplicación.

En el **capítulo 8**, encontraremos de manera detallada toda la información económica y sobre plazos de desarrollo asociados al proyecto, se explicará cada una de las subtaréas a realizar par la consecución de los objetivos y el personal necesario para estas.

Debido a la naturaleza de las aplicaciones móviles , nuestro afán de mejora y la identificación de nuevas necesidades que se nos planteen a lo largo del desarrollo, ya sean de ideas propias de la aplicación o la posibilidad de crecer en otros mercados, en el **capítulo 9**, se barajarán las posibles futuras expansiones, desarrollos o implementaciones que puedan hacer llegar nuestra aplicación a un nivel superior de calidad o de acercamiento a usuarios.

Finalmente, cerramos la memoria con la inclusión de las **referencias**, los **apéndices** en donde se ofrecen las licencias sobre las que se basa este proyecto , un breve esquema sobre las diferentes **cuentas** y servicios asociados necesarios para el correcto funcionamiento de la aplicación y una **iconografía** donde se reflejan los diferentes logos e iconos pertenecientes a la aplicación

CAPÍTULO 2

ESTADO DEL ARTE

2. ESTADO DEL ARTE

En este capítulo vamos a hacer un repaso a la situación actual del entorno de las tecnologías móviles y más concretamente de los SmartPhones. Estudiaremos como ha sido la evolución del mercado y las tecnologías hasta llegar a día de hoy, deteniéndonos en los acontecimientos y las tecnologías más importantes.

Haremos un resumen de las diferentes soluciones que existen en el mercado, los diferentes sistemas operativos, así como sus virtudes y defectos enfocándonos en explicar no sólo el estado del arte actual, si no de establecer un marco sólido que nos ayude a comprender el porqué de este proyecto y las tecnologías empleadas.

2.1. El Smartphone, historia y tecnología.

Smartphone: *nombre masculino*

“Teléfono celular con pantalla táctil, que permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.”¹

Aunque comúnmente es habitual pensar que el primer Smartphone de la historia es el desarrollado por Apple, bajo el nombre comercial I-phone, la realidad es muy anterior.

En los orígenes de los dispositivos electrónicos personales, las compañías luchaban por abrirse un hueco en el mercado mediante los llamados PDA (por sus siglas en inglés Personal Digital Assitant), que venía a ser una agenda electrónica a la que se le fueron sumando cada vez más capacidades de un pc de la época².

Paso a detallar los modelos que han marcado una mayor importancia en la evolución y posteriormente se adjunta una infografía algo más extensa en cuanto a modelos.

IBM SIMON

El primer dispositivo al que podemos llamar Smartphone es el IBM SIMON^{3, 4, 5 y 6}. Este terminal hizo su primera presentación el 23 de noviembre de 1992 en la Feria Comdex, Las Vegas, bajo el nombre de proyecto “Angler”, se combinó un teléfono celular y el PDA⁷ en un

solo dispositivo, lo que permite al usuario hacer y recibir llamadas telefónicas agregando funcionalidades tales como calendario con citas, agenda, calculadora, reloj mundial, bloc de notas electrónico, anotaciones manuscritas a mano alzada...

Asistentes y la prensa de la época mostraron un notable interés en el dispositivo por lo que IBM junto a BellSouth se lanzaron a crear una versión para ser llevada al mercado final, tras diversos retrasos en 1994 bajo el nombre comercial "Simon Personal Communicator" se hace efectiva su comercialización.

Este dispositivo contaba con una pantalla táctil monocromática de 4 pulgadas, teclado qwerty, procesador de 16 bits a 16mhz del fabricante Intel basado en la arquitectura x-86. En el apartado de almacenamiento contaba con 1MB tanto de memoria RAM como de almacenamiento interno, siendo notable para la época.



Ilustración 1: Historia, IBM Simón

La alianza de BellSouth e IBM consiguió unas ventas de 50.000 dispositivos, cifra reducida debido a su elevado coste de venta al público y su limitada duración de la batería.

Como curiosidad, este dispositivo no fue considerado un "smartPhone" hasta la aparición del siguiente prototipo.

Ericsson GS88

En 1997 aparece el termino "Smartphone" de la mano de la presentación del Ericsson GS88 ⁸ que era el nombre comercial del desarrollo "Penélope". Este terminal corría bajo el Sistema operativo de 16 bit GEOS de GeoWorks ya disponía además de las funcionalidades comunes a la época, navegación web, un teclado QWERTY físico, como la serie "communicator de los modelos Nokia de la época (aunque estos no disponían de navegador), modo avión, puerto infrarrojo, conexión a PC mediante cables serie, manos libres y modem integrado.

Jamás llegó a producirse a escala y sólo se quedó en un prototipo con apenas 200 unidades fabricadas, pero fue importante pues supuso la base de desarrollo para el siguiente modelo.



Ilustración 2 - Historia, Ericsson GS88

Ericsson R380

El Ericsson R380 ^{9 y 10} fue el primer terminal apodado “smartPhone” que realmente llego al gran público. Su salida al mercado se produjo en el año 2000, contaba con una pantalla monocroma resistiva, un potente hardware Con 2 MB de RAM, 4 de ROM, pantalla de 3,5 pulgadas, resolución de 120 x 360 píxeles y un diseño y peso que se asemejaba perfectamente a los móviles de la época. Uno de sus avances más importantes fue la incorporación de Ericsson en el proyecto Symbian y este fue el primer dispositivo en incorporar dicho sistema operativo.



Ilustración 3 Historia, Ericsson R380

BlackBerry 850 ¹¹ y 5810 ¹²

En 1999 había sacado el modelo 850 que fue revolucionario por su capacidad de gestionar el correo electrónico de manera inalámbrica y su teclado completo. En el 2002 a estas capacidades se le suma que fue el primer smartPhone de la marca con capacidad completa para GSM y GPRS , su gran pantalla y el sistema operativo BlackberryOs aportaban ventajas que la llevaron a los primeros puestos en el ranking de ventas. Esta marca, se hace muy relevante a la hora de hablar de smartPhones pues aún teniendo un sistema operativo propietario, aceptaba la comunicación con otros muchos sistemas, lo que la hacía muy adaptable, a su vez incorporaba servicios de mensajería instantánea basados en red, lo cual se ha podido demostrar el alcance posterior que ha supuesto.



Ilustración 4, Historia, BlackBerry 850



BlackBerry 5810

IPHONE:

En 2007 se produce un acontecimiento que revolucionaría la industria y marcaría la manera de entender el Smartphone hasta nuestros días, aparece el Iphone ^{13, 14, 15}. Fue una revolución pues era el primer teléfono en incorporar multitud de capacidades multimedia, una potente cámara de 2mp, acelerómetro, y una pantalla a todo color completamente interactiva y multitáctil, lo que provocaba la desaparición completa de los teclados y botones. Esta característica unida a que el terminal utiliza el Operativo OS-X, con todas las capacidades que esto permite genera una facilidad de uso sin precedentes que lo catapultan rápidamente a ser un superventas y marcar un antes y un después en la vida de los smartPhones.



Ilustración 5, Historia, Iphone 1

Su pantalla de 3,5" capacitiva contaba con una resolución de 320x480 px. Iba acompañado de un procesador Samsung, 128Mb de RAM y 4 GB de almacenamiento.

HTC DREAM:

El Htc Dream ^{16, 17, 18} , salió a la venta en Octubre de 2008, fue el primer teléfono inteligente que incorporó Android como Sistema Operativo, contaba con pantalla táctil, un teclado qwerty físico, pues no disponía de teclado en pantalla. Como principales novedades la inclusión de las Google Apps y Android Market.

Este dispositivo contaba con pantalla de 3,2 pulgadas, Procesador Qualcomm, 192MB de RAM, Cámara de 3,15Mp , GPS y una amplia conectividad HSPA.



Ilustración 6- Historia, HTC Dream

MOTOROLA DROID ¹⁹:

Este terminal es importante, no tanto por sus innovaciones pues mantiene teclado qwerty de sus predecesores, aunque agrega la posibilidad de teclado en pantalla (agregado en Android 1.1) ni por su avanzada tecnología, pantalla de 3,7 wxga, potente procesador de Texas Instruments Basado en ARM, sus 256mb de memoria RAM o la posibilidad de utilizar SD externas como medio de almacenamiento. Destaca principalmente por ser el primer éxito comercial de Android en referencia al gran público. La agresiva campaña que Google presentó en conjunto con Verizon permitió a este dispositivo con Android 2.0 llegar a 1 millón de unidades en apenas 74 días.



Ilustración 7- Historia Motorola Droid

Llegados a este Punto, nos encontramos con la instauración de lo que sería la semilla para establecer la situación actual. En donde una vez los dispositivos basados en Android e los han ido ganando cuota de mercado hasta casi hacer marginal las cuotas de sus competidores

Aunque existan terminales basados en Windows Phone, Bada, Firefox Os... por cuotas de mercado existen dos grandes familias de sistemas.

Dispositivos Apple ¹⁵

Desde su lanzamiento en 2007, el I-phone ha ido evolucionando a través de 6 generaciones que lo han ido dotando de mayores capacidades, potencia y añadidos, aunque la esencia y funcionalidad sigue pareja al diseño original.

Modelo	iPhone	iPhone 3G	iPhone 3GS	iPhone 4	iPhone 4s	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6 Plus
Sistema inicial	OS 1.0	OS 2.0	OS 3.0	iOS 4.2.5	iOS 5.0	iOS 6.0	iOS 7.0		iOS 8.0	
Pantalla	89 mm (3,5") cristal LCD 480 × 320					102 mm (4,0") 960 × 640 px a 326 PPP		119 mm (4,7") 1334 x 750, 326 ppp	140 mm (5,5")1920 x 1080 px 401 ppp	
Memoria	4, 8 o 16 GB	8 o 16 GB	8, 16 o 32 GB		8, 16, 32 o 64 GB	16, 32 o 64	8, 16 o 32	16, 32 o 64 GB	16, 64 o 128 GB	
Dimensiones	115 mm H 61 mm W 11.6 mm D	115.5 mm H 62.1 mm W 12.3 mm D	115.2 mm H 58.6 mm W 9.3 mm D		123.8 mm H 58.6 mm W 7.6 mm D	124.4 mm H 59.2 mm W 8.97 mm D	123.8 mm H 58.6 mm W 7.6 mm D	138.1 mm H 67.0 mm W 6.9 mm D	158.1 mm H 77.8 mm W 7.1 mm D	
GPU	PowerVR MBX Lite 3D		PowerVR SGX535 GPU		PowerVR SGX543MP2	PowerVR SGX543MP3		PowerVR G6430	PowerVR GX6450	
CPU	620 MHz Samsung 32-bit RISC		Samsung	Apple A4(SoC)	1 GHz doble Apple A5	1.3 GHz doble Apple A6		1.3 GHz doble 64-bit Apple A7	1.4 GHz doble 64-bit Apple A8	
RAM	128 MB DRAM		256 MBDRAM	512 MB DRAM		1 GB (LPDDR2 en 5 y 5C; LPDDR3 en el resto)				
Cámara	2 MP f/2.8		3 MP	5 MP	8 MP			8 MP	8 MP	
Lanzamiento	junio de 2007	11 julio 2008	junio 2009	junio 2010	octubre 2011	sept 2012	sept 2013	septiembre de 2013	septiembre de 2014	

Ilustración 8 Historia, Tabla Dispositivos Apple

Dispositivos Basados en Android:

Android, al tratarse de un sistema de libre distribución, ha sido adoptado a multitud de dispositivos de diferentes fabricantes con un amplísimo rango de especificaciones y características. A nivel de ventas, sobresale la familia galaxy ^{20, 21} ofertada por Samsung, por lo que un brevísimo resumen de estos dispositivos nos ayuda a conceptualizar la idea general del progreso que ha sufrido esta plataforma.

Modelo	Galaxy ²²	Galaxy S2 ²³	Galaxy S3 ²⁴	Galaxy S4 ²⁵	Galaxy S5 ²⁶	Galaxy S 6 ²⁷	Galaxy S6 Edge ²⁷
Sistema inicial	Android 2.1/2.2	Android 2.3	Android 4.0	Android 4.2.2	Android 4.4.2	Android 5.0.2	
Pantalla	Pantalla táctil Super AMOLED de 4.3" (480x800 píxeles)	Súper AMOLED Plus de 4,27" (480x800 píxeles)	HD Súper AMOLED de 4,8" (1280 x 720 píxeles)	Súper AMOLED 5.0" 1080 x 1920 pixel	Súper AMOLED 5.1" 1080 x 1920 pixels	Súper AMOLED 1440 x 2560 pixels, 5.1 pulgadas	Súper AMOLED 1440 x 2560 pixels, 5.1 pulgadas, Curva en laterales.
Memoria	8 ó 16 GB	16GB	16/32GB	16/32/64GB	16/32GB	32/64/128GB	
Dimensiones	122,4 mm x 64,2 mm x 9,9 mm	125,3 x 66,1 x 8,49 mm.	136,6 x 70,6 x 8,6 mm.	136.6 x 69.8 x 7.9 mm	142 x 72.5 x 8.1 mm	143.4 x 70.5 x 6.8 mm	
GPU	PowerVR SGX 540	MALI-400	Mali-400	PowerVR SGX544MP3	Adreno 330	Mali-T760	
CPU	Samsung S5PC110	Exynos dual core a 1,2 GHz.	Exynos 4412 (quad-core) a 1.4 GHz	Exynos 5 Octa 5410	Qualcomm Snapdragon 801	Exynos 7420 octa-core 64 bits	
RAM	512MB	1GB	1GB	2 GB	2GB	3GB	
Cámara	5MP	8MP	8MP	13MP	16MP	16MP	
Lanzamiento	2009	Mayo 2011	Mayo 2012	Abril 2013	Febrero 2014	Febrero 2015	

Ilustración 9 – Historia, Tabla Dispositivos Samsung



Sources
<http://9to5google.com/2014/02/25/samsung-illustrates-the-evolution-of-its-galaxy-s-series-in-a-new-infographic/>
<https://s-media-cache-ak0.pinimg.com/736x/52/5d/78/525d78b0404927b907acb05e617e836.jpg>
<http://www.businessinsider.com/mind-blowing-facts-about-samsung-2013-4?top=1>

mobile
MADHOUSE
 mobilemadhouse.co.uk



Ilustración 10 – Historia, Infografía Samsung

Fuente:

<http://www.mobilemadhouse.co.uk/Content/Images/uploaded/Blog%20Images/Samsung-Galaxy-S-Infographic.png>

A modo ilustrativo agrego una infografía que detalla de manera visual los hitos expuestos:



Ilustración 11 - Historia,
Infografía SmartPhones 1



Ilustración 12 - Historia, Infografía SmartPhones 2



Los Años 2010 y 2011 fueron una verdadera revolución en cuanto a variedad de fabricantes y modelos.

Este Boom ha marcado la tendencia hasta la actualidad haciendo cada vez dispositivos con mejores prestaciones en ciclos más cortos de desarrollo.

Por otro lado, no se encuentra la aparición de ideas revolucionarias o cambios profundos que estén marcando una revolución distinta en la actualidad a la iniciada en estos años, entre el 2011 y el 2015, fecha de realización del proyecto, el mercado a tendido a una completa explotación y mejora de estas tecnologías. No obstante, si realizáramos este estudio en el futuro, es seguro que encontrásemos algún producto que aporte un toque diferenciador que marque tendencia en años venideros.



IMAGEN:

<http://www.bitrebels.com/technology/the-evolution-of-smartphones-infographic/>

Ilustración 13 - Historia, Infografía SmartPhones 3

2.2. Hardware

A nivel técnico estos dispositivos están centrados en ofrecer la mayor potencia posible pero teniendo unas restricciones en cuanto a consumo muy limitadas intrínsecas a su naturaleza como dispositivo móvil.

2.2.1. CPU & GPU

CPU:

Aunque existen otras arquitecturas la que se ha impuesto en la totalidad de los dispositivos móviles ha sido la Basada en ARM ^{28, 29}.

ARM es una arquitectura basada en instrucciones RISC, las cuales al ser más cortas y sencillas de ejecutar que otras arquitecturas como X-86 (CISC), aportan los principales elementos que la hacen idónea para su aplicación en dispositivos móviles.

- Arquitectura RISC
- Un ajustado consumo energético.
- Bajo coste de diseño y fabricación.
- Elevada eficiencia por watio.
- Escalable.
- Licenciable mediante acuerdos con ARM HOLDING

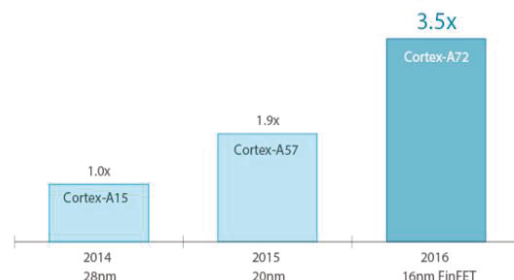
La empresa encargada de realizar la arquitectura, ARM HOLDINGS (Británica) establece los modelos de cada generación. Es fácil buscar en la Wikipedia y encontrarnos desde la primera especificación la ARMv1 hasta la actual ARMv8, dentro de las cuales existen diferentes núcleos de proceso que estarán optimizados según el uso que se les vaya a dar, rendimiento, consumo...

ARM realiza el establecimiento de la tecnología y sus pautas. Estas pueden seguirse como hacen multitud de fabricantes, rockchip, renesas, marvell... o sobre los que pueden trabajar y realizar mejoras, como es el caso de Qualcomm con sus Snapdragon, Samsung con Exynos, Nvidia Tegra, Los chips de Mediatek y un largo etc.



Actualmente las CPUS más potentes son las basadas en el diseño de referencia de ARM en su núcleo Cortex A-57 y A-53 (snapdragon810, mediatek Helios x-20) estando ya cercana la liberación de las especificaciones del Cortex A-72.

Increase in sustained performance in smartphone power budget



GPU:

El tratamiento al que se enfrenta la GPU es totalmente distinto al de una CPU, al tener que hacer multitud de veces una tarea más específica, la fuerza bruta y la multiplicidad de cores predomina en este campo.



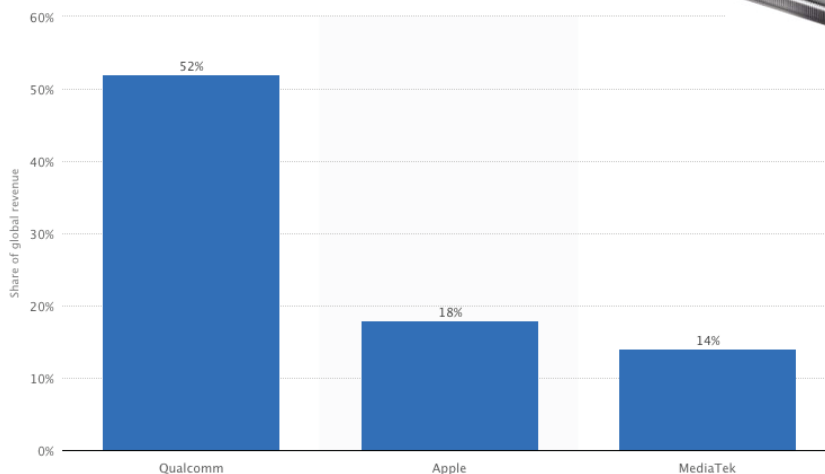
Existen muchos tipos de arquitecturas dependiendo del fabricante, por ejemplo, Nvidia en sus procesadores Tegra para móviles utiliza su experiencia con su arquitecturas Maxwell y Kepler que tan buenos resultados le ha dado en pc y optimiza al máximo los procesos de fabricación y frecuencias para mantener unos consumos ajustados.

PowerVR es una empresa que posee una tecnología única llamada "tile rendering"³⁰ que consiste en pintar únicamente lo que el usuario ve en pantalla, con lo que con un hardware menos potente y con menor consumo consigue los resultados de otras soluciones más potentes ofrecidas por otros fabricantes.



Qualcomm por su lado, cuenta con un desarrollo propio denominado Adreno el cual proviene de la división de AMD de los chips de bajo consumo desarrollados por la absorbida ATI y que fue vendida a Qualcomm en 2009 ³¹

Para finalizar, al igual que con las CPU, ARM tiene su propio desarrollo, las GPU MALI que cualquier fabricante puede licenciar y fabricar.



**Ilustración 15 E.A-
Porcentaje Mercado
por Fabricante chips**

Porcentaje de mercado por fabricante.

Fuente: <http://www.statista.com/statistics/233415/global-market-share-of-applications-processor-suppliers/>

2.2.2. hardware complementario

A día de hoy, un smartPhone no sólo se caracteriza por tener un hardware capaz de hacer millones de operaciones por segundo, el concepto de smartPhone no sería completo si a esa pieza de ingeniería no se le agregarán tecnologías que lo dotan de una usabilidad y unas capacidades que lo separan completamente del mundo PC.



La inclusión de sistemas **GPS** que posicionan con gran exactitud el dispositivo, permitiendo a los desarrolladores utilizar esta característica para hacer herramientas basadas en la localización, como pueden ser Navegadores, redes sociales Geo localizadas, sistemas de seguimiento del dispositivo...

El sistema GPS es muy demandante de energía, por lo que está apoyado por el sistema A-GPS que basa su posicionamiento con respecto a las antenas de telefonía, menor precisión gran ahorro de energía, según la finalidad de la aplicación el desarrollador optará por activar uno u otro medio.

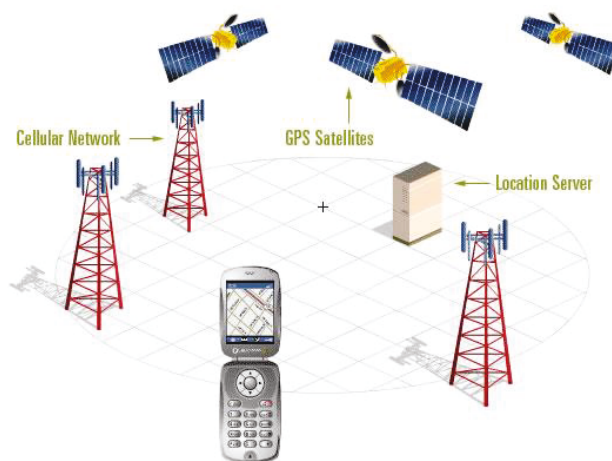
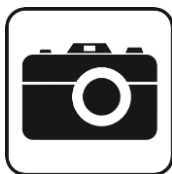


Ilustración 16 E.A- Detalle funcionamiento A-GPS



La Cámara, otro elemento indispensable del smartPhone, ya no sólo son los megapíxel, los fabricantes cada vez integran mejores ópticas y características de entorno más profesionales. No es raro ver cámaras con amplia apertura focal, estabilizadores ópticos, capaces de capturar altas tasas de fotogramas por segundo, etc.



El **Bluetooth**, sistema de comunicación inalámbrica de corto alcance, aplicado sobre todo en dispositivos manos libres, reproductores, SmartWatches y otros gadgets que se conecten a nuestro terminal y se encuentren en un entorno cercano a este.



NFC: por sus siglas “Near Friend Communication”, es otro es un sistema de comunicación de ultra corto alcance, en donde para poder vincular los dispositivos la distancia tiene que ser como máximo de unos pocos centímetros. Aunque en Europa no está muy implantada en Países como Japón ya es un medio habitual para realizar pagos o un ejemplo más sencillo, utilizar el Smartphone como tarjeta de transporte público en vez de tener que tener la tarjeta como en España, que viene a ser un chip NFC al que se le programan los datos del usuario.

2.2.3. Sensores

Conjuntamente a los diferentes agregados ya mentados, se encuentran multitud de sensores³³ que sirven de apoyo o complemento a otros sistemas y que sin los cuales las funcionalidades y sobre todo la experiencia de usuario se vería mermada enormemente.

- Acelerómetro
- Giroscopio
- Magnetómetro
- Sensor de Profundidad
- Sensor de Luz
- Barómetro
- Termómetro
- Sensor de Humedad Relativa
- Podómetro
- Sensor de ritmo Cardíaco
- Sensor de Huellas Digitales

Dependiendo del Terminal, fecha de fabricación y gama en la que nos encontremos, existirán unos u otros dentro del dispositivo, no obstante existen multitud de gadgets que están basados en estos sensores y que permiten mediante software de terceros agregar funcionalidades a cualquier dispositivo.

2.2.4. Conectividad a internet

La definición de Smartphone detalla claramente que es un dispositivo “que permite al usuario conectarse a internet” Con lo que es importante saber las diferentes comunicaciones que puede tener nuestro smartPhone hacia este medio, sobre todo porque dependiendo del tipo de conexión que este utilice pueden ser necesarias ciertas precauciones a la hora de programar una aplicación.



WIFI: Los smartPhones, casi desde sus orígenes, cuentan con conectividad Wifi, esto les permite conectarse a redes existentes y muy extendidas por todo el mundo.

Estas redes tienen como ventaja principal que comúnmente suelen ser redes no saturadas con bajas latencias y altas tasas de transferencia, en donde la posibilidad de perder un paquete que genere un error en nuestra aplicación es prácticamente nula.

Desde el punto de vista del programador es la situación óptima de conectividad, pues la estabilidad y los tiempos de respuesta suelen ser óptimos y no requieren de mecanismos de control extras a los ofrecidos por los sistemas operativos.



Redes Móviles: Una de las principales ventajas de los SmartPhones es la capacidad de utilizar las redes móviles para conectarse a internet. Esto les asegura su conectividad.

Las Redes móviles han evolucionado aumentando sus capacidades, ampliando su cobertura e implantando tecnologías que permiten una mayor transmisión de datos. Ha día de hoy conviven 3 de las 4 generaciones, la primera ya esta obsoleta pues sólo servía para transmitir voz y se basaba en un modelo analógico (antigua Teleline en España). El Uso de estas redes suele suponer un mayor tiempo de

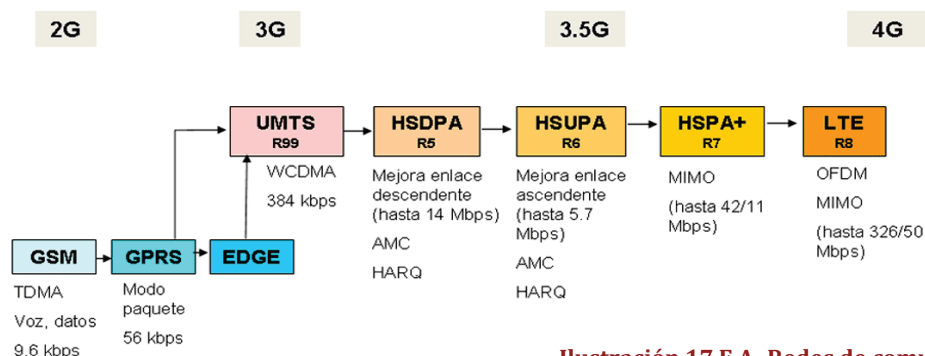


Ilustración 17 E.A- Redes de comunicación

respuesta que obligan a el programador a adoptar medidas como bloqueos de la interfaz, mensajes informativos de proceso u otros que aseguren el correcto funcionamiento del software.

2.3. Sistemas Operativos

La utilidad de un Sistema operativo en el móvil es la misma que en cualquier otro dispositivo, gestionar los recursos Hardware y proveer de los servicios necesarios a el software, a los programas de aplicación que sobre él se ejecuten.

2.3.1. Android

La historia de Android ^{34,35,36} se remonta a 2003, momento en el que fue fundada la Android inc. el cual y como curiosidad, debe su nombre a la novela de Philip K.Dick, ¿sueñan los androides con ser ovejas eléctricas?, la cual seguramente conoceremos más por su adaptación cinematográfica “Blade Runner”

Este Sistema Operativo esta basado en un núcleo Linux y sus principales características de diseño son:

- Núcleo Monolítico³⁷: la totalidad del espacio del sistema operativo trabaja a nivel de núcleo.
- Interfaz gráfica.
- Soporta las Plataformas: ARM, X86, Mips y Power PC
- Licencia apache y GNU GPL (kernel)

Android contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario.



Ilustración 18 E.A- Detalle arquitectura Capas Android

La arquitectura ³⁸ de capas de abstracción de Android , se detallará cada una de las capas en el capítulo 3.

Desde su Aparición han ido surgiendo versiones y mejoras del Sistema operativo, detallamos a continuación su cronología con las versiones que supusieron un mayor avance tanto por despliegue como por innovaciones.



Ilustración 19 E.A- Detalle Versiones Android

Se funda Android Inc.	Octubre 2003	
	Agosto 2005	Google Compra Android inc.
Se crea Android “unveiled”	5 Noviembre 2007	
	12 noviembre 2007	Sale el primer SDK
Sale al mercado el HTC Dream	23 Sept 2008	
	9 Feb 2009	Aparece la versión Android 1.1
ANDROID 1.6 “DONUT”	15 Septiembre 2009	
<ul style="list-style-type: none"> ✓ Nueva cámara ✓ Mejora de búsqueda por voz y búsqueda ✓ Mejora compatibilidad con pantallas grandes 		
	26 Octubre 2009	
		ANDROID 2.0 “ECLAIR”
		<ul style="list-style-type: none"> ✓ Mejoras en UI ✓ Mejoras en Contactos y sincronización ✓ Agrega soporte para Exchange ✓ Agrega soporte Multicuenta ✓ Mejora cámara ✓ Nueva versión teclado en pantalla ✓ Nuevo navegador HTML 5 ✓ Mejora velocidad ✓ Mejoras en Calendar y Google Maps ✓ Bluetooth 2.1

ANDROID 2.2 FROYO

20 Mayo 2010

- ✓ Se agregan controles a la cámara
- ✓ Tethering con hasta 8 Hotspots
- ✓ Teclado Multilenguaje
- ✓ Mejoras de Performance
- ✓ Mejoras en el bluetooth, llamadas de voz, compartir contactos...
- ✓ Nueva Home



6 Diciembre 2010



ANDROID 2.3 GINGERBREAD

- ✓ Refinamiento de la UI, más rápida y precisa
- ✓ Mejora de la introducción por teclado
- ✓ Mejoras control de energía
- ✓ Se añade capacidad de matar las aplicaciones
- ✓ Llamadas a través de Voip
- ✓ NFC
- ✓ Agregado un Download Manager
- ✓ Soporte multi-cámara
- ✓ Soporte para Barómetro, Giroscopio y acelerador lineal.

ANDROID 3.0 "HONEYCOMB"

22 Febrero 2011

- ✓ Se agrega soporte para Tablet
- ✓ Rediseño del teclado
- ✓ Se agrega la opción de copiar & pegar
- ✓ Soporte para teclados físicos
- ✓ Se agrega soporte para pantallas más grandes
- ✓ Soporte Multi-core
- ✓ Agrega librerías de alto rendimiento para 2d y 3d.



10 Mayo 2011



ANDROID 3.1

- ✓ Se refina la UI con animaciones
- ✓ Se incluye soporte para periféricos USB
- ✓ Se agregan pantallas escalables
- ✓ Soporte para joystick y Gamepads
- ✓ Mejora del sistema WIFI
- ✓ Actualizaciones apps de correo, contactos, calendario...
- ✓ Se agrega soporte a funcionalidades de empresa

ANDROID 3.2

18 Julio 2011

- ✓ Mejoras para Tablet
- ✓ Compatibilidad para apps de tamaño fijo
- ✓ Acceso directo a la SD desde las apps de terceros
- ✓ Mejora de la Api para controlas múltiples pantallas.



19 Octubre 2011



ANDROID 4.0 ICE CREAM SANDWICH

- ✓ Se agrega "aplicaciones Recientes"
- ✓ Se Refina la UI
- ✓ Carpeta Home y Favoritos
- ✓ Acciones de bloqueo de pantalla
- ✓ Función "quick response" para llamadas
- ✓ Widgets escalables
- ✓ Wifi Direct
- ✓ Desbloqueo por reconocimiento facial
- ✓ NFC Sharing
- ✓ Control de los Datos de Red

- ANDROID 4.1 "JELLY BEAN"**
- ✓ Mejoras en el sistema táctil y transiciones
 - ✓ Notificaciones expandibles y de múltiples tamaños
 - ✓ Teclado adaptativo
 - ✓ Audio USB
 - ✓ Google Wallet
 - ✓ Review Instantánea de la foto

9 Julio 2012



29 Octubre 2012

ANDROID 4.2

- ✓ Multiusuario en Tablet
- ✓ Poto Sphere 360º
- ✓ Teclado Swype nativo
- ✓ Mejoras en sistema de captura de cámara

- ANDROID 4.4 "KIT-KAT"**
- ✓ Importantes mejoras en el consumo y optimización de la memoria RAM
 - ✓ Chromium Engine al Webview
 - ✓ Restricciones de acceso en disco a las aplicaciones y otras mejoras de seguridad
 - ✓ Bluetooth Message
 - ✓ Audio Tunelling
 - ✓ Capacidad de imprimir a través de wifi
 - ✓ Se hace pública la API de mensajería
 - ✓ Nueva Api de acceso a SD y memoria interna

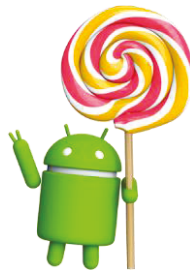
3 Octubre 2013



3 Noviembre 2014

ANDROID 5.0 LOLLIPOP

- ✓ Soporte previsualización de impresión
- ✓ Project VOLTA, mejoras en la duración de la batería
- ✓ OPENGL 3,1, AEP
- ✓ INPUT / OUTPUT de audio a través del USB
- ✓ Se separan los updates del Webview a los del sistema.
- ✓ "tap and Go" nueva gestión de cuentas de usuario
- ✓ añade posibilidad de "usuario invitado"
- ✓ soporte para CPU de 64 Bits
- ✓ Mejoras profundas en la máquina virtual, reemplazando Dalvik por una mucho más avanzada.



- ANDROID 5.1**
- ✓ Soporte Cultísima Nativo
 - ✓ **Device Protection en caso de robo**
 - ✓ **Cambia la prioridad del sistema de notificaciones**

9 Marzo 2015

21 Abril 2015

ANDROID 5.1.1

- ✓ Arregla un preocupante Bug de Memoria existente desde la Versión 5.0

Ilustración 20 E.A- Tabla detallada Versiones Android

Uno de los principales retos a los que se enfrenta Android es su elevada fragmentación. Esta situación se produce porque cuando una nueva versión es creada, los fabricantes tienen que adaptarlo a sus dispositivos con sus personalizaciones, a su vez, cada operadora deberá igualmente trabajar sobre la actualización para bloquear, si procede los terminales de manera restrictiva a sus redes.

Esta situación nos genera un mercado donde conviven multitud de versiones diferentes, lo que supone un coste añadido al desarrollador a la hora de programar. Google, consciente de estas limitaciones pone a disposición las denominadas “**support Library**” en donde en mayor o menor medida, se van agregando todas las nuevas funcionalidades para que sistemas más antiguos puedan ejecutar correctamente cualquier aplicación.

Esta decisión de utilizar o no esta support library siempre será por parte del programador y no todas las nuevas funcionalidades estarán disponibles, pues muchas veces existen limitaciones hardware que hacen inviable agregar ciertas competencias.

FRAGMENTACIÓN TEMPORAL

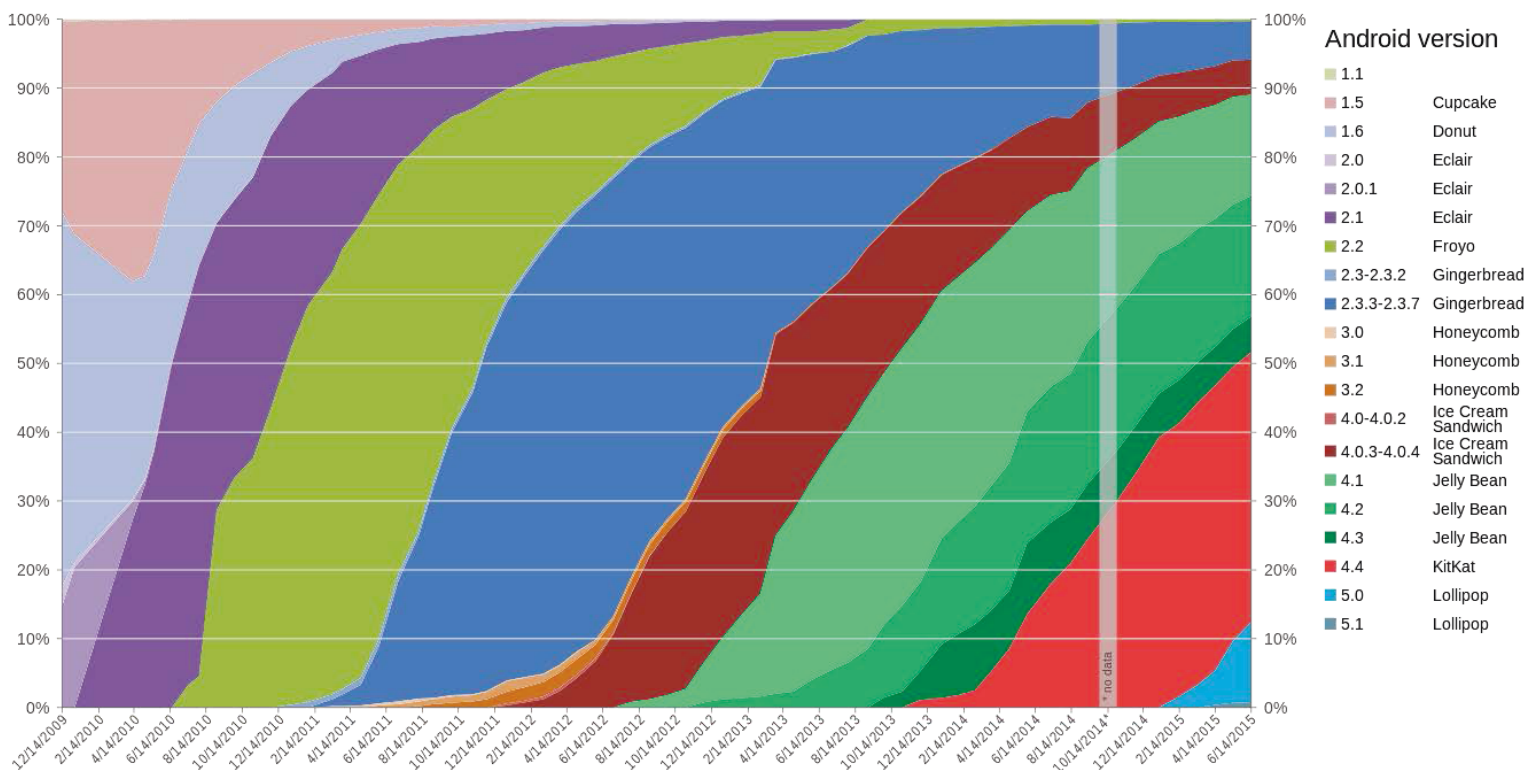
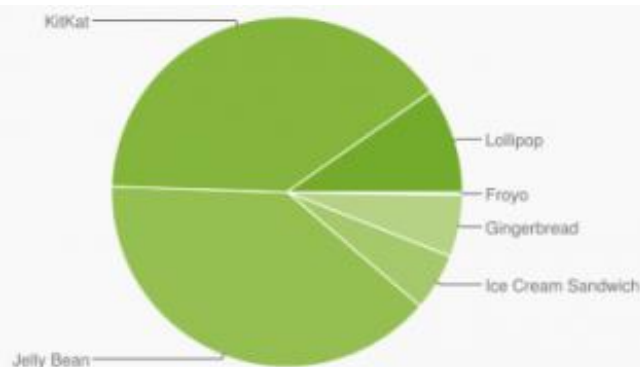


Ilustración 21 E.A- Fragmentación Temporal Dispositivos Android

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.3%
4.1.x	Jelly Bean	16	15.6%
4.2.x		17	18.1%
4.3		18	5.5%
4.4	KitKat	19	39.8%
5.0	Lollipop	21	9.0%
5.1		22	0.7%



Data collected during a 7-day period ending on May 4, 2015.
Any versions with less than 0.1% distribution are not shown.

Ilustración 22 E.A- Detalle Fragmentación Android Mayo 2015

En este gráfico podemos ver la fragmentación en Mayo de 2015.

Fuente: Google.

2.3.2. iOS

El segundo sistema más extendido del mercado es el encargado de hacer funcionar todos los terminales de la marca Apple, Recibe el nombre de iPhone OS, o iPhone OSX o Simplemente iOS ^{39, 40, 41, 42}

Su presentación se hizo el 1 de Junio del 2007 conjuntamente a el iPhone original, esta Basado en el anterior Sistema Operativo “Darwin” en desarrollo desde el año 2001, el cual a su vez fue continuación del desarrollo NextStep de 1989. Esta basado en UNIX y sus principales características son:

- Núcleo Híbrido XNU ⁴³: es un híbrido entre monolítico y micro-núcleo que intenta obtener lo mejor de cada una, el paso de mensajes de los micronúcleos y la velocidad de los núcleos monolíticos para desempeñar ciertas tareas.
- Soporta las Plataformas: ARM, X86, X86-64 y Power-PC
- Software Libre mediante Apple Public Source License.
- Basado en el concepto directo de manipulación mediante entrada táctil

El sistema operativo de iPhone facilita marcos para la incorporación de tecnologías, servicios y características en las aplicaciones para iPhone. El sistema operativo esta compuesto por cuatro capas⁴⁴ de abstracción.



Ilustración 23 E.A- Detalle Capas S.O. Ios

Capa del núcleo del sistema operativo:

Es la capa más baja del Sistema operativo, el núcleo, situándose directamente con el manejo del hardware. Es el responsable de todos los aspectos del S.O. lleva el control de la memoria virtual, gestión de procesos, los Threads, Sistema de archivos etc....

Al ser un núcleo híbrido también cuenta con un conjunto de interfaces, basadas en lenguaje C, que permiten el acceso de bajo nivel y proveen soporte para Red, "bonjour", Asignación de memoria, cálculos matemáticos...

Capa del núcleo de servicios Core "Services layer":

Proporciona acceso a los servicios principales del SO de iPhone con los marcos de la capa del núcleo de servicios. Entre las principales tecnologías disponibles a alto nivel de esta capa encontramos: i-cloud, Conteo de referencias automáticas (ARC), block Objects, Grand Central Dispatch (GCD), Sq.Lite & CoreData y soporte XML

Capa media:

Es la encargada de generar el contenido multimedia disponible en el móvil.

- | | |
|-------------|---|
| Tecnologías | ➤ Core Grafics |
| Gráficas: | ➤ Core Animation |
| | ➤ Open GL / GLKit |
| | ➤ Core Text |
| | ➤ Interfaces de E/S formatos de imagen |
| | ➤ Aseets |
| Sonido: | ➤ Media Player |
| | ➤ Framework AV |
| | ➤ OpenAL |
| | ➤ Core Audio |
| Video: | ➤ Media Player |
| | ➤ Las clases de UiimageController |
| | ➤ Core Media |
| | ➤ Framework AV |
| Air Play | ➤ Integración con Apple TV |
| | ➤ Control de flujos de Sonido Inalámbricos con dispositivos compatibles |

Capa Cocoa Touch

Cocoa Touch es la capa más importante para el desarrollo en iOS. Posee los Frameworks que proporciona el API Cocoa.

Está formada por dos Frameworks fundamentales:

- UIKit: contiene todas las clases que se necesitan para el desarrollo de una interfaz de usuario.
- Foundation Framework: define las clases básicas, acceso y manejo de objetos, servicios del sistema operativo.

Apple siempre, casi siempre ha desarrollado una nueva versión de su sistema con el lanzamiento de sus terminales.

IPHONE OS 1

Junto con el iPhone Original, y según la revista Time se consideró el invento del Año.

- Revolucionaria interfaz multitáctil
- Aplicaciones de Correo / navegador revolucionarios
- Ui con animaciones
- Teclado táctil 100% funcional

Junio 2007



Junio 2008

iPhone 3G



IPHONE OS 2

Se presenta junto a el Iphone 3g

- Se presenta la App Store
- Soporte de apps nativas de terceros
- Visualización de documentos Office
- Reproducción de Videos desde Safari
- Geoetiquetado de fotografías
- Sincronización Push
- Mejoras en la seguridad y nuevos estándares

IPHONE OS3

Se presenta en conjunto con el Iphone 3GS

- Se agrega la función de corta&pegar
- Mensajes MMS
- Mejoras en el GPS
- Grabación de video
- Mejoras en el software de cámara
- HTML 5
- Control por voz
- Notificaciones Push de terceros.

Junio 2009

iPhone 3G S



Junio 2010



iPhone 4



IOS 4

Se presenta junto a el Iphone 4

- Se presentan más de 100 nuevas características
- 1500 nuevas Apis
- Multitarea
- FaceTime
- Game Center
- Soporte teclados Bluetooth
- Fotografías HDR
- Gran esfuerzo en soporte a grandes empresas
- iMovie
- iBook
- Carpetas

2.

- IOS 5**
- Junto con el iPhone 4S
- Centro de notificaciones
 - iMessage
 - iCloud
 - Siri
 - Integración con Twitter
 - Sincronización inalámbrica con iTunes
 - En total más de 200 nuevas funcionalidades.



- IOS 6**
- Se presenta junto a el Iphone 5
- Nuevo Servicio de Mapas
 - Integración con Facebook
 - Compartir Fotos en streaming
 - Passbook
 - Mejoras en Safari
 - Fotografías panorámicas
 - Rediseño de multitud de componentes
 - Mejoras en la seguridad
 - Nuevas API

- IOS7**
- Junto con el iPhone 5S y 5C
- TouchId
 - Mejoras en el software de cámara
 - Mejoras en multitasking
 - AirDrop
 - FaceTime con Audio
 - Mejoras en la interfaz, cambiándola a colores más planos con gradientes y transparencias



- IOS 8**
- Se presenta junto a el Iphone 6 y 6Plus
- Apple Pay
 - Apps de salud nativa
 - iCloud-Drive
 - Llamadas Wifi
 - Family Sharing
 - NFC
 - I-watch
 - Soporte para teclados de terceros
 - Mejoras en la Api
 - Nuevas funcionalidades de mensajería

- IOS9**
- El 8 de junio se presentan sus principales características, aunque todavía no hay fecha de disponibilidad.
- Mejoras notables en Siri
 - Centrado en Mejorar el rendimiento y la seguridad
 - Api Spotlight
 - Notas a mano alzada
 - Mapas con tráfico y transporte
 - Apple News
 - Mejoras específicas para i-pad



Ilustración 24 E.A- Tabla Versiones Ios

Los dispositivos Apple, a diferencia de Android, las actualizaciones dependen directamente de Apple y ninguna empresa intermedia tiene que tocar el software, puesto que es un único hardware para el único software. Esto provoca que el nivel de fragmentación de los dispositivos sea prácticamente inexistente, estando el límite en la compatibilidad de los sistemas más antiguos con los nuevos requisitos del sistema operativo. Actualmente desde el iPhone 4s en adelante todos los dispositivos son actualizables a la última versión.

A modo comparativo entre las dos principales plataformas del mercado, se adjunta gráfico con la salida de sus dispositivos de referencia y versiones del sistema operativo.

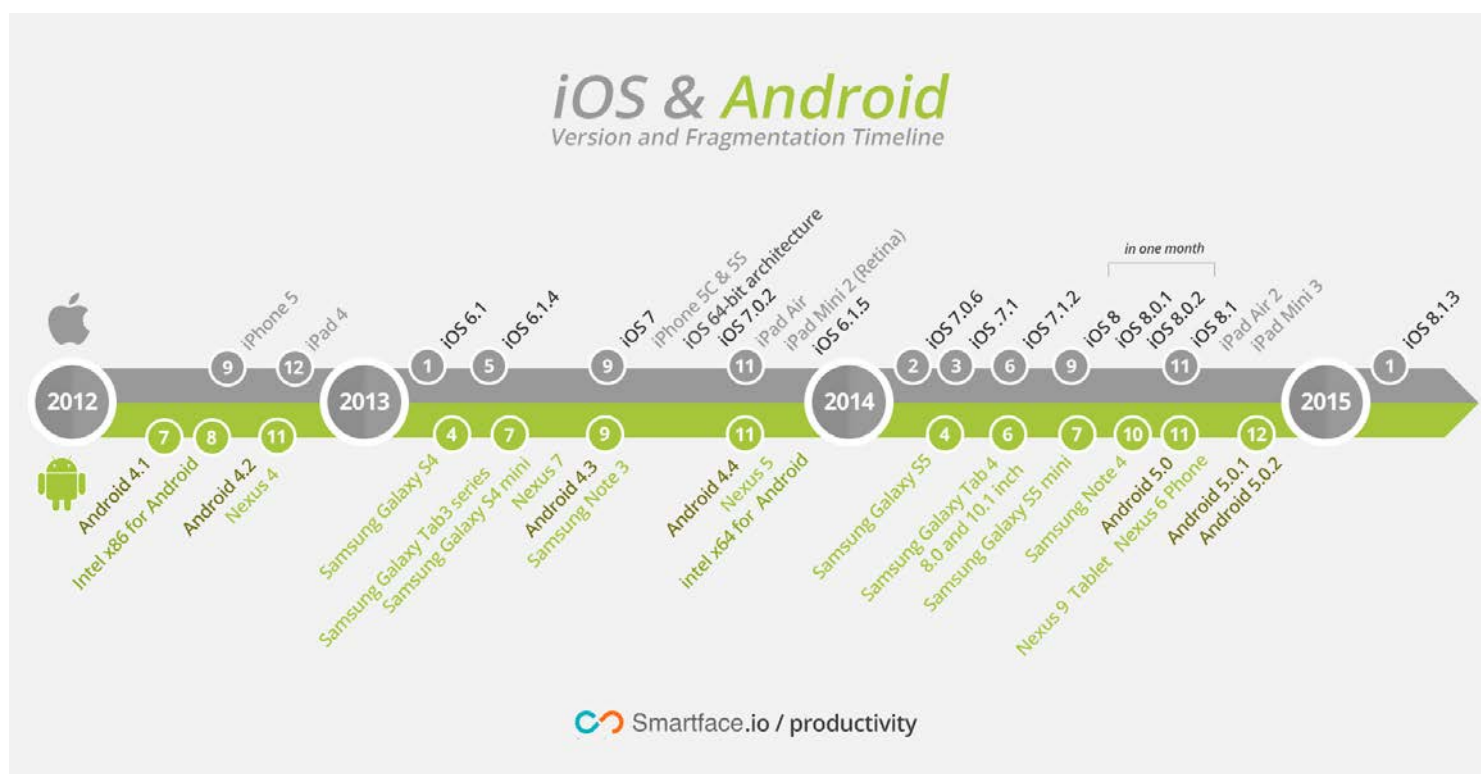


Ilustración 25 E.A- Línea Histórica comparativa versiones Android / Ios

2.3.3. Windows Phone & Windows 10

Microsoft, también tiene su solución para dispositivos móviles, Windows Phone ⁴⁵. Esta versión del sistema operativo es sucesora de Windows Mobile y a diferencia de este, esta enfocado al mercado de consumo en vez de al empresarial. De todos es el único sistema del que conocemos su fecha de caducidad por lo menos de su nombre comercial, tras su presentación al mercado Windows 10 pasará a copar todo el espectro de venta de Microsoft.

Este sistema operativo a sufrido cambios profundos en cada una de sus versiones por lo que debemos hacer una diferenciación por versiones.

Windows Phone 7.X	Windows Phone 8.X	Windows 10
<ul style="list-style-type: none">➤ Basado en Windows CE➤ Núcleo Monolítico➤ Soporta X86, x86-64, Mips, ARM y SuperH➤ Navegador Explorer➤ Actualización por Cable a PC	<ul style="list-style-type: none">➤ Basado en Windows NT➤ Núcleo híbrido➤ Soporta X86, x86-64y ARM➤ Interfaz Modern UI➤ Navegador Explorer➤ Actualización autónoma	<ul style="list-style-type: none">➤ Basado en Windows NT➤ Núcleo híbrido➤ Soporta X86, x86-64y ARM➤ Interfaz Modern UI➤ Navegador Edge➤ Actualización autónoma

Ilustración 26 E.A- tabla comparativa Versiones Windows Phone



Basándonos en los últimos datos ⁴⁶ La cuota de mercado en España de Windows phone esta en torno al 2,5% y su único representante se queda en los terminales fabricados por Nokia, empresa que previamente ha comprado Microsoft y su familia Lumia.

Otro de los puntos diferenciadores de Windows phone es su plena integración con las consola Xbox y el uso de las herramientas más utilizadas en ofimática, Microsoft office sobre el que además implementa los servicios OneDrive y Sharepoint permitiendo trabajar de manera colaborativa y/o desde varios entornos.



BlackBerry ⁴⁷ cuenta actualmente con una cuota de mercado de el 0,3%, su Sistema operativo esta enfocado plenamente al uso profesional debido a su alta seguridad

y amplia compatibilidad con sistemas empresariales como Lotus, Novell o Exchange. El Sistema Operativo en sí esta basado en QNX , Sistema Unix micronúcleos que se empezó a desarrollar en 1982.

Soporta las plataformas BlackBerry De Alpha y la futura línea de teléfono inteligentes BlackBerry, basados en chips ARM.

Cuenta con el mercado de aplicaciones propietario “BlackBerry world”.



FirefoxOs ^{49,50} Es un Sistema Operativo de código Abierto desarrollado por Mozilla Corporation basado en un núcleo Linux.

Este sistema tiene la peculiaridad de que esta basado en la utilización de HTML, CSS y Javascript como su lenguaje base para el desarrollo de aplicaciones, lo que lo haría idóneo para los desarrolladores.

Se compone de una arquitectura separada en 3 componentes:

1. **Gonk**: Conformado por el kernel Linux y una capa de abstracción de hardware.
2. **Gecko**: El entorno de ejecución.
3. **Gaia**: La interfaz gráfica de usuario.

Al igual que sus competidores cuenta con su propia tienda de aplicaciones denominada “Firefox MarketPlace”



Ubuntu Touch es un Sistema Operativo de núcleo Linux desarrollado por Canonical, su intención es crear un sistema con una interfaz que pueda utilizarse en ordenadores de sobremesa, portátiles, netbooks, Tablets y teléfonos inteligentes. Como este sistema tiene la capacidad de ejecutar cualquier aplicación que corriera en la distribución Ubuntu de escritorio. Algunas de sus características más destacadas son:

- Pantalla de inicio sin sistema de bloqueo/desbloqueo (funciona con un sistema de gestos).
- Ubuntu Touch incluye como aplicaciones centrales de medios sociales y medios de comunicación (por ejemplo, aplicaciones de Facebook, YouTube, y un lector de RSS). Lo que lo hace un S.O. eminentemente social.
- Integración con Ubuntu One.

Ubuntu touch, cuenta con una tienda propia de aplicaciones, aunque su número de apps es todavía muy reducido y como principal inconveniente es su acceso, sólo puedes entrar a la tienda de aplicaciones desde el terminal con el S.O. instalado, no existe web ni otras vías de acceso.



Tizen ^{51, 52, 53} es un sistema Operativo que actualmente está siendo desarrollado por Samsung tras haber abandonado su proyecto anterior BADA. Tizen es una mezcla de los sistemas Meego y bada. Su núcleo está basado en Linux, monolítico. Es compatible con los sistemas ARM y X86 y al igual que Firefox OS, la capa gráfica está desarrollada en HTML5. Tizen tiene la ventaja de ser un sistema que se ha mostrado muy ligero y además tiene la capacidad de ejecutar aplicaciones de Android, con lo que lo hace muy interesante. De momento, el único terminal que lo monta es el Samsung Z1, aunque según Samsung, sus planes son que para el 2020 sea una realidad en competencia contra Google/iOS.



Symbian, fue un S.O. desarrollado mediante la colaboración de Nokia, Sony Mobile Communications, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. EL objetivo era crear un potente sistema capaz de competir con Palm o Microsoft, que eran los que poseían mayor cuota de mercado cuando se inició su desarrollo, Poco a poco fue perdiendo interés y Nokia se convirtió en su único dueño, sus dispositivos fueron los últimos en montar este S.O. discontinuado en octubre de 2012.

2.3.5. Comparativa de mercado de los principales Sistemas operativos

Empezaremos viendo las cuotas de Mercado ⁴⁸. Se observa la supremacía clara del binomio Android & Ios.

Periodo	Android	iOS	Windows Phone	BlackBerry OS	Otros
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Ilustración 27 E.A- Tabla Cuota Mercado S.O.

	Apple iOS 8	Android 5.0	Windows Phone 8	BlackBerry 10	Firefox OS 2.2
Compañía	Apple	Open Handset Alliance	Microsoft	BlackBerry	Mozilla Foundation
Núcleo del SO	Mac OS X	Linux	Windows NT	QNX	Linux
Licencia de software	Propietaria	Libre y abierto	Propietaria	Propietaria	Libre y abierto
Año de lanzamiento	2007	2008	2010	1999	2013
Fabricante único	Sí	No	No	Sí	No
Variedad de dispositivos	Modelo único	Muy alta	Media	Baja	Muy baja
Soporte memoria externa	No	Sí	Sí	Sí	Sí
Motor del navegador web	WebKit	WebKit/Chromium (Blink)	Trident	WebKit	WebKit
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry World	Firefox Marketplace
Número de aplicaciones	800.000 (marzo 2013)	800.000 (marzo 2013)	130.000 (enero 2013)	100.000 (enero 2013)	¿?
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	Sin coste	Sin coste
Otras tiendas sin supervisión	No	Si	No	Si	Si
Familia CPU soportada	ARM	ARM, MIPS, x86	ARM	ARM	ARM, x86
Soporte 64 bits	Si	Si	No	No	No
Máquina virtual	No	Dalvik / ART	.net	No	Navegador Web
Lenguaje de programación	Objective-C, C++	Java, C++	C#, Visual Basic, C++	C, C++, Java	HTML5, CSS, JavaScript
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Multiusuario	No	Si	No	No	No
Modo invitado	Si	Si	No	No	No

Ilustración 28 E.A- Tabla resumen Diferencias S.O.

Fuente: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/98-comparativa-con-otras-plataformas>

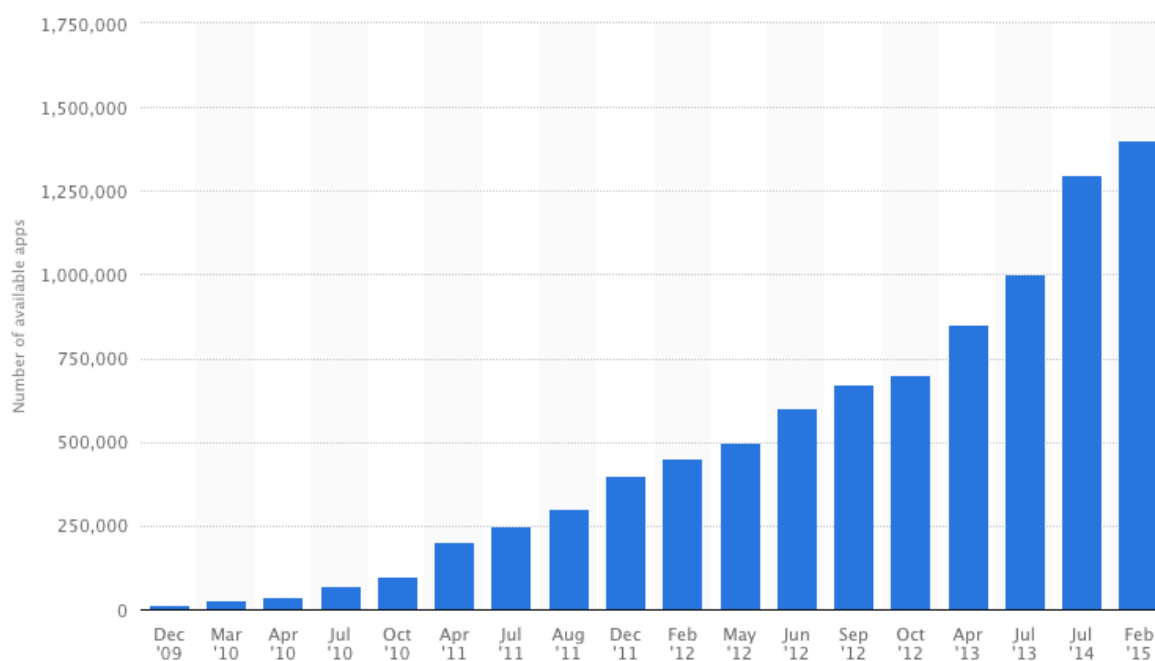
2.4. Los diferentes mercados de Aplicaciones

En este apartado se realiza una comparativa sobre los diferentes mercados de aplicaciones. Empezamos con un resumen que nos ayude a posicionarnos, estudiando la incidencia que podría tener ese mercado en nuestro proyecto.



Google play, es el resultado de google al unificar el antiguo Android Market con Google Music. Es la tienda oficial de los dispositivos que corren Android como S.O.

Dispone de subsecciones específicas para Juegos, Kiosco, Películas y música y cuenta con aplicaciones tanto gratuitas como de pago, para lo cual utiliza la plataforma propia Google Wallet. Observando su gráfico de aplicaciones disponibles podemos ver como la aceptación de los dispositivos Android lo hace crecer año tras año .



Fuente:<http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

Ilustración 29 E.A-Gráfico Aplicaciones Disponibles Google Play



Apple, cuenta con su propia tienda de aplicaciones llamada Appstore. No fue la primera en salir al mercado, pero su rápido crecimiento la permitió en tan sólo 3 meses colocarse como la principal tienda de apps hasta este año, momento en que ha sido relegada a un segundo puesto. Es compatible solamente con los dispositivos Apple y de también permite apps tanto gratuitas como d pago.

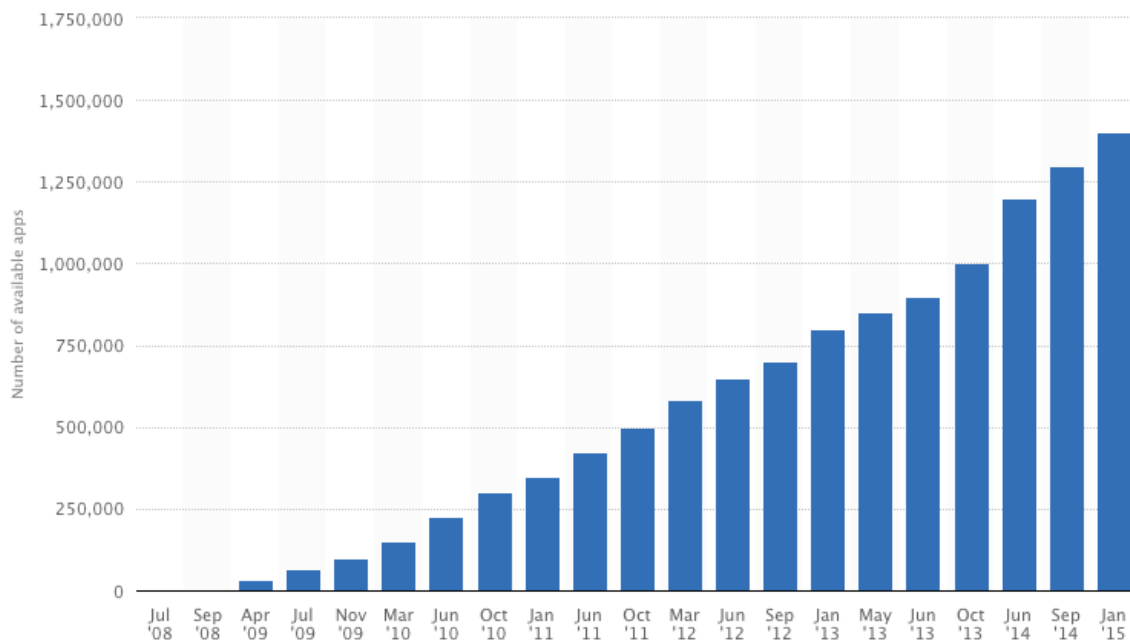


Ilustración 30 E.A- Gráfico aplicaciones Disponible AppStore

Fuente:<http://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>

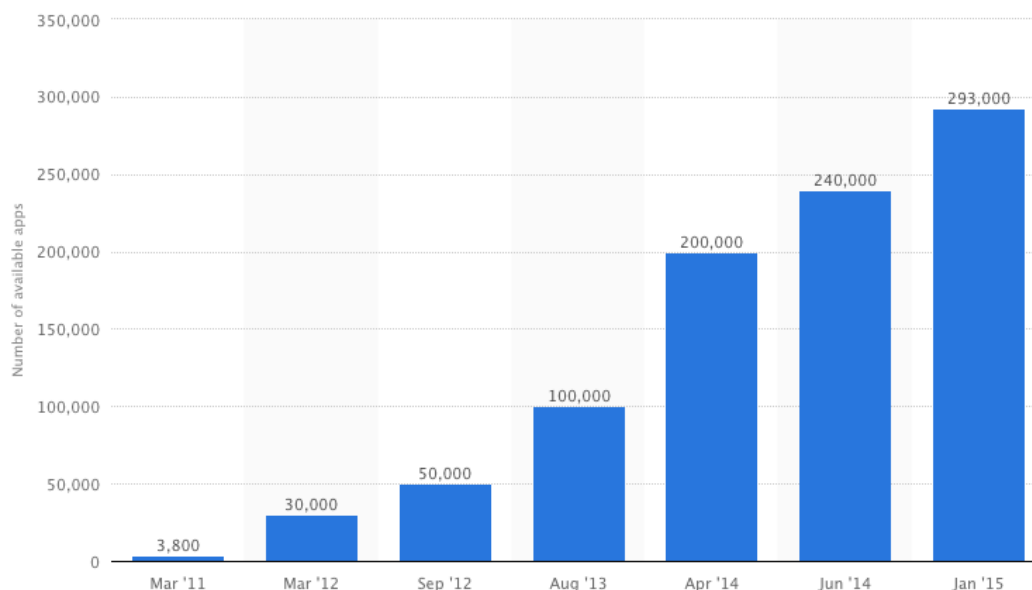


La plataforma desplegada por BlackBerry para sus dispositivos recibe el nombre de BlackBerry world, su primera versión se lanzó el 1 de abril de 2009. Soporta apps tanto gratuitas como de pago y es gratuita para los desarrolladores. Actualmente la plataforma BlackBerry esta creciendo en número de aplicaciones pero sufre una acusada disminución de descargas.



Amazon Appstore ^{54, 55} Es una tienda de aplicaciones para el sistema operativo Android. Fue lanzada al mercado en Marzo del 2011 su

crecimiento se vio reforzado en septiembre de ese mismo año con el lanzamiento de una Tablet propia de Amazon, “kindle Fire” en donde se substituyo el mercado de aplicaciones de Android por el de Amazon. La política que mantiene de cara a los desarrolladores es sencilla, se queda un 30% , no obstante a día de hoy es la Store más lucrativa para los desarrolladores.



www.statista.com/statistics/307330/number-of-available-apps-in-the-amazon-appstore/

Ilustración 31 E.A- Detalle Aplicaciones Amazon AppStore



La Tienda de Microsoft se llama Windows Phone Store ⁵⁶ Como todo lo que rodea a la nueva plataforma de Microsoft esta basado en MetroUI, representando todo mediante “celdas”. Microsoft lanzo la primera tienda en octubre de 2010, Esta organizada en 16 categorías principales y 25 subcategorías.

Para los desarrolladores, deben pagar un canon de 99\$ Anuales y pueden elegir diseñar para un dispositivo en concreto.

En Marzo de 2015 la Windows Phone Store llego a las 385.000 apps, su crecimiento es constante, pero la oferta es muy limitada si la comparamos con sus principales competidores.

2.5. Consideraciones específicas para HISPANIA UC3M

En este apartado vamos a explicar porque hemos optado por la plataforma Android como plataforma de desarrollo para llevar a cabo este proyecto. Para ello, nos vamos a nutrir del marco establecido hasta el momento, por los apartados 2.1 a 2.4

Tenemos que tener en cuenta que uno de nuestros propósitos iniciales es conseguir que nuestra aplicación llegue a el mayor número potencial de personas posible, por ello, gran parte de las elecciones que aquí se describen se hacen persiguiendo esa idea.

2.5.1. Entorno Android

La elección del sistema Operativo y por lo tanto la plataforma sobre la que crear nuestro primer desarrollo ha sido sencilla. Android es claramente nuestra mejor opción debido a:

- ✓ Es la plataforma con mayor número de usuarios potenciales
- ✓ Es compatible con multitud de terminales
- ✓ Nos permite un bajo costo de desarrollo
- ✓ No tenemos que pagar nada por su desarrollo
- ✓ El hardware disponible es lo suficientemente maduro como para que nuestra app sea compatible con terminales más antiguos.
- ✓ Existen multitud de herramientas en la red compatibles con el sistema
- ✓ La compatibilidad con los sistemas de backend utilizados es nativa
- ✓ Las herramientas de desarrollo son Open Source y bien documentadas
- ✓ Es un sistema estable y sobradamente probado
- ✓ No Necesitamos más que un Pc convencional para realizar su desarrollo.

Se establece como versión mínima para ejecutar la aplicación la versión 4.0 , Ice-Cream Sandwich, de Android. Este requisito lo ponemos puesto que versiones anteriores tienen limitaciones con las Api de terceros utilizadas, tanto de backend como de conectividad con las redes sociales. A su vez el obligarnos a tener una compatibilidad a las versiones 2.X de Android nos limita las capacidades de diseño de nuestra interfaz.

Esta decisión solamente deja fuera a menos de un 6% de los dispositivos, los cuales, presumiblemente y en una gran mayoría pensamos que son de gente que hace un uso muy limitado del terminal y no pasarían a ser parte de nuestros usuarios.

2.5.2. Hardware necesario

La aplicación no hace uso de un Hardware específico ni necesita acceso a tecnologías gráficas o multimedia de uso intensivo. Esto es un requisito deseable pues nuestro cometido es llegar a la mayor cantidad de público posible, una limitación técnica nos cerraría cuota de posible mercado.

Cualquier dispositivo que cuente con la versión 4 instalada del Sistema operativo será capaz de ejecutar de una manera satisfactoria nuestra aplicación, independientemente de su antigüedad.

2.5.3. Google Play

El proyecto Hispania Uc3m no busca un mercado de nicho, busca llegar al público general por lo que nos centraremos en las tiendas más representativas en cuanto a volumen de descargas y usuarios únicos potenciales.

Tras la elección de nuestra plataforma, podemos elegir entre dos mercados de distribución Google play y Amazon Store. Para nuestra primera fase de proyecto optamos por Google play debido a:

- ✓ Es la Store con mayor número de usuarios
- ✓ Esta instalada en todos los terminales Android
- ✓ Por el pago de una pequeña cuota inicial (19\$) nos da acceso a todas las funcionalidades
- ✓ La plataforma es madura y los usuarios se fían de ella
- ✓ La operativa de control y aceptación sobre las apps es rápida y sencilla
- ✓ Existen multitud de herramientas para hacer seguimiento y estadísticas.

2.6. Diseño de la Aplicación Móvil

Actualmente el diseño es un aspecto totalmente clave de una aplicación, en un mercado abarrotado de contenido audiovisual y conjunto a un usuario que cada vez interactúa más con multitud de aplicaciones de grandes desarrolladores que cuentan con potentes equipos, es fundamental contar con un diseño que no provoque el rechazo del usuario, ya sea o bien porque esta mal estructurado o sencillamente porque no sea visualmente atractivo.

Desde la aparición de las primeras aplicaciones, los desarrolladores apoyados por los diseñadores han explotado miles de ideas en un esfuerzo de comprensión hacia este nuevo medio, pues supone la interactividad directa táctil sobre unos reducidos y múltiples tamaños de pantalla.

Actualmente lo que se persigue en el desarrollo de aplicaciones es que, aunque no proporcionen un verdadero valor real, todos conocemos multitud de aplicaciones que podíamos denominar de baja calidad, pero ofrezcan una experiencia de usuario satisfactoria.

Me gustaría citar al escritor Douglas K.von Duyne cuando en sus libros de patrones Web dice “Cuando los visitantes coinciden en evaluar con alta puntuación el contenido, facilidad de uso, rendimiento, fiabilidad y satisfacción, se denomina sitio Web centrado en el usuario” pues lo considero aplicable a nuestro contexto de aplicación y debemos buscar la máxima de Diseño centrado en Usuario.

2.6.1. Elementos Clave de la interfaz basada en usuario.

✓ **Facilidad de uso:**

La experiencia de usuario varía enormemente entre diferentes personas, es por ello que se hace necesario que la aplicación sea tan sencilla de usar como sea posible sin perjudicar su funcionalidad.

- Reducir el número de opciones, establecer jerarquías por uso y quitar distracciones que puedan confundir al usuario.
- El texto debe ser claro, fuentes claras y sencillas con tamaños que permitan su fácil legibilidad, este es un punto crítico en Android, pues su fragmentación nos puede jugar malas pasadas en algunos dispositivos.
- Los procesos de interacción con el usuario han de ser simples. Tanto la intuición como los estudios que lo evidencian nos demuestran que cuantas más veces tenga que pulsar el usuario para llegar a un sitio determinado, menor es la probabilidad de que este sea visitado, o si es un proceso, completado. Se debe minimizar el nº de pasos.
- No permitir que el usuario se desubique, los diseños deben dar señales y pistas que ayuden al usuario a identificar claramente en que punto de la app se encuentra.

✓ **Satisfacción:**

La medida en que una aplicación este adecuada a el objetivo que persigue es proporcional a la satisfacción que esta genera en el usuario. Para entender lo importante que es este elemento, sólo tenemos que pensar en la última aplicación que hayamos descargado y nos haya sorprendido positivamente, ya sea por su calidad visual o por la eficacia de sus objetivos.

✓ **Rendimiento:**

El rendimiento que debe aportar una aplicación no debe entenderse únicamente como el tiempo que tarda en cargar o ejecutar alguna rutina de descarga o similar, sino a la a la implementación de los procesos en los que el usuario tenga que interaccionar con el sistema . Se procurará una navegación sencilla y fluida, y en caso de que haya que establecer esperas, estas deberán de ser claras por su cometido y ser minimizadas siempre que sea posible.

✓ **Contenido:**

La información y el cometido para los que un usuario ha descargado la aplicación debe ser claro, sin sobrecargar con datos innecesarios.

2.6.2. Usabilidad de la aplicación

A día de hoy, es fundamental para un desarrollador conocer los principio básicos de la arquitectura , usabilidad e interfaz de usuario. El desarrollador realizará una organización de la estructura de datos clarificando lo complejo, un mapa de información que permita al usuario encontrar sus caminos personales a la operatividad con el sistema. Deberá clarificar las funcionalidades y la visión de la interfaz equilibrando las necesidades de información y navegación. Por último es importante prever como crecerá nuestra aplicación, pues en el proyecto que se desarrolla con el tiempo aumentarán las partidas y los amigos que un usuario posee y esto genera un mayo volumen de datos sobre nuestra interfaz.

En este punto cito a Jakob Nielsen, considerado de los mayores especialistas en usabilidad, en su obra 'Law of the Web experience': "Los usuarios pasan la mayor parte de su tiempo en otros sitios". Esta afirmación es totalmente verídica, ya se trate de sitios web o de aplicaciones móviles. Las convenciones establecidas por las aplicaciones más extendidas no deberán ser muy diferentes de las que establezcamos en nuestro propio desarrollo.

Se utilizarán términos y patrones conocidos por los consumidores evitando nuevas acepciones o comportamientos que puedan confundir al usuario.

2.7. Aplicaciones Móviles: Juegos, educación y entretenimiento.

Llegados a este punto vamos a estudiar diferentes aplicaciones que debido a su naturaleza y éxito comercial son buenos ejemplos sobre los que basar nuestro desarrollo. Este estudio es fundamental por lo expuesto previamente en cuanto a la usabilidad, no debemos “reinventar la rueda” y nos ayudará a identificar diseños y requisitos que son esperables por parte del usuario en una aplicación de esta naturaleza.

2.7.1. Logos Quiz

Me gusta Hablar de Logos Quiz, pues la aplicación original, es la app Española más descargada de la historia, con más de 40 millones de descargas en todo el mundo en la Appstore y 34 millones en Google Play. Además he tenido la suerte de poder conocer personalmente a Javier Pérez , un joven pamplonés que junto a su hermano Fundó AticoD y que estuvo a punto de morir de éxito tras poner por tiempo limitado su app de forma gratuita en el Appstore.



Ilustración 32 E.A- Logo, LogosQuiz

Su historia también tiene un carácter agri dulce, pues la pasividad con la que hicieron su migración a Android permitió a competidores copiar su aplicación y perder todo el boom inicial que ellos mismos habían generado. No obstante fueron capaces de remontar esta situación mediante la inclusión del modo multijugador y actualmente se encuentran en el top 2, siendo el primer puesto para los desarrolladores que primero hicieron la copia.

Aunque con el paso de los años la app ha ido aumentando en complejidad, lo que me gusta de ella es la sencillez de su sistema de juego. **El sistema de juego de Hispania UC3M esta basado en los conceptos básicos de la primera versión de Logos Quiz**, obviamente adaptados a la temática y con cambios que se adecuan a la estructura de preguntas en texto.

En la actualidad ha Logos Quiz a cambiado el sistema de juego, pues se a incorporado un sistema de monedas. En su origen, en Logos Quiz aparecen una serie de logos sobre los que hay que escribir el nombre de la marca a la cual pertenecen, y esto amigos, es la clave de todo su éxito. Accesible a todos y una mecánica de juego totalmente trivial.

El sistema pasa de las marcas más generales en sus niveles más sencillos a otras menos generalistas según vas avanzando. En cada nivel la complejidad aumenta debido a que el número de logos es mayor y mayores las restricciones para pasarlo.

El sistema de respuesta es de palabra única, esta situación nos es muy positiva pues de forma inherente hace la lógica de datos sobre la temática del juego sencilla, con tres campos recreamos la lógica de juego; nivel, imagen de marca y solución. A su vez, esta sencillez nos permite olvidarnos del sistema primario de juego y centrar los recursos en la gamificación del juego.



Ilustración 33 E.A- Detalles LogosQuiz

2.7.2. QuizUp



Ilustración 34 E.A-
Logo QuizUp

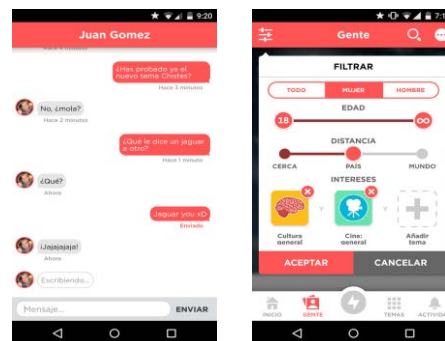
QuizUp es una aplicación Android que descubrí por casualidad un día en el metro de Madrid, observando un chico que tenía sentado a mi lado e iba jugando. Me llamó la atención la cantidad de preguntas que fallaba, sobre todo porque existía un gran número de ellas sobre las que pensaba que la respuesta era muy obvia. En este momento me di cuenta de cuán incultos somos, yo incluido y lo sensato de orientar nuevos desarrollos en pro de generar una sociedad más culta.

Lo más interesante de esta aplicación es su contenido educativo indirecto, pues de una manera totalmente gamificada entre jugadores aleatorios y siguiendo un sistema de preguntas y respuestas en las que el usuario por ocio intenta ganar al adversario para progresar en un ranking, va adquiriendo una serie de conocimientos de una manera interactiva.

De QuizUp, me quedo con la idea de entretenimiento educativo y el sistema de tiempo para proponer una solución, pues la aplicación a parte consta de muchas partes como selección de temática, chats, foros... que extrapolan la envergadura inicial del proyecto sobre el que se está trabajando.



<-El sistema de juego consiste en contestar lo más rápidamente posible y dentro de un tiempo establecido una serie de preguntas.



QuizUp También consta de chat, Filtro de oponentes...

Ilustración 35 y 36 Detalle QuizUp

2.7.3. Pasopalabra



Esta aplicación no es ni muy conocida ni excesivamente descargada de Google play, pues cuenta con un intervalo de instalaciones de entre 100k-500k, me gustaría mentarla brevemente pues de ella saque la condición de poner un tiempo máximo de respuesta.

La forma de implementarlo ha sido distinto pues aquí el tiempo que se da al usuario es para completar una tanta de palabras, una con cada letra del abecedario y en el proyecto el tiempo se ha implementado por cada palabra.

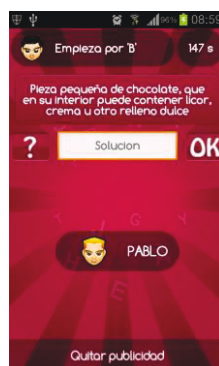
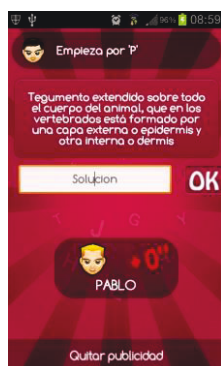


Ilustración 37, 38 y39 Detalle Pasopalabra

2.7.4. El Juego de las Adivinanzas



Este juego, al igual que el anterior ha llegado a mis manos a través de mis sobrinos, de él me gusto el sistema que tiene para adivinar las palabras, el cual te delimita la longitud máxima de la palabra y en mi caso, se ponen algunas letras de ejemplo.

La decisión de realizar esta implementación en donde se pinta en pantalla de una manera dinámica la longitud de la palabra, acarree ciertos problemas posteriores a la hora de programarlo, pues debido a la fragmentación de Android, según el largo de la palabra había algunas que no cabían o se descuadraban los guiones sobre los que se sitúan las letras.

Este sencillo juego ofrece al usuario un ancho delimitado de palabra, el conjunto de letras desordenadas que contienen y una adivinanza que la cual la solución es la palabra pedida.



Ilustración 40 Detalle Adivinanzas

CAPÍTULO 3

ANDROID EN PROFUNDIDAD

3. ANDROID

Tras haber definido en el apartado anterior un breve marco de la plataforma Android, así como de su evolución a través de sus versiones desde su aparición hasta nuestros días, en este apartado vamos a realizar un estudio profundo a su arquitectura y los principales aspectos que tiene el desarrollo de una aplicación en esta plataforma.



3.1. Arquitectura

En el apartado 2.3.1 se vieron las principales características del Sistema Operativo, en este punto estudiaremos las entrañas que tiene Android de una manera detallada.

A modo resumen recordemos Android, Sistema Operativo basado en Linux, de núcleo monolítico con una arquitectura jerarquizada en 4 Capas

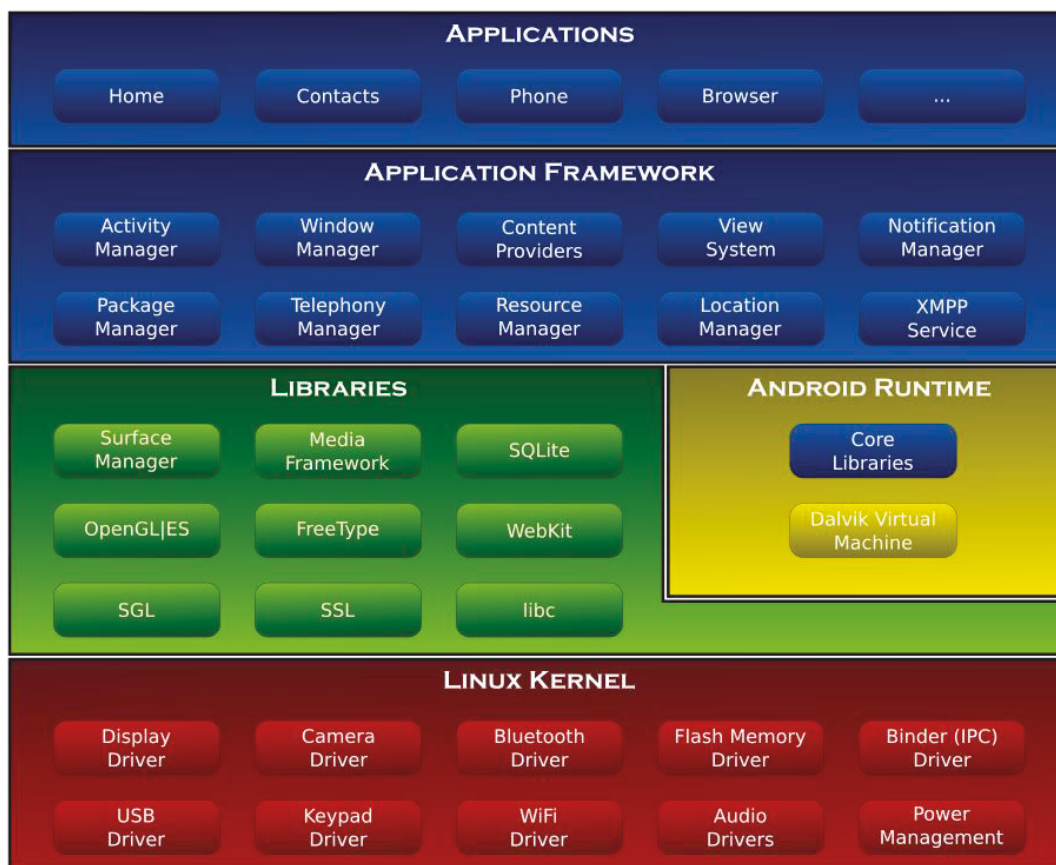


Ilustración 41
Diseño Capas
Android

IMG Wikipedia, info curso Android UC3M

3.1.1. El núcleo Linux

Basado en el núcleo de Linux 2.6 como una capa de abstracción para el hardware disponible en los dispositivos móviles. Contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes.

Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

3.1.2. Android Runtime

Esta Capa, también llamada “tiempo de ejecución de Android” está al mismo nivel que las librerías de Android. Constituyen las Core Libraries, que son librerías con multitud de clases Java.

En esta capa se encuentra la Máquina virtual de ejecución Android. Esta basado en la máquina virtual utilizada en Java, pero dadas a las limitaciones Hardware de los dispositivos se tuvo que implementar una maquina virtual más ligera, **DALVIK**.

Dalvik consigue su cometido de eficiencia gracias a que esta basada en el uso de registros, a diferencia de la maquina Java estándar que usa Pilas. Es una máquina virtual interprete para la que se crearon los archivos .dex⁵⁹, de formato optimizado para ser eficiente y mapeable en memoria. En el se contienen múltiples clases y diseño enfocado a la reutilización de datos. Su objetivo fundamental es el de cualquier máquina virtual, permitir el compilado del código a bytecode independiente de la máquina de ejecución para posteriormente interpretar este bytecode en la ejecución.

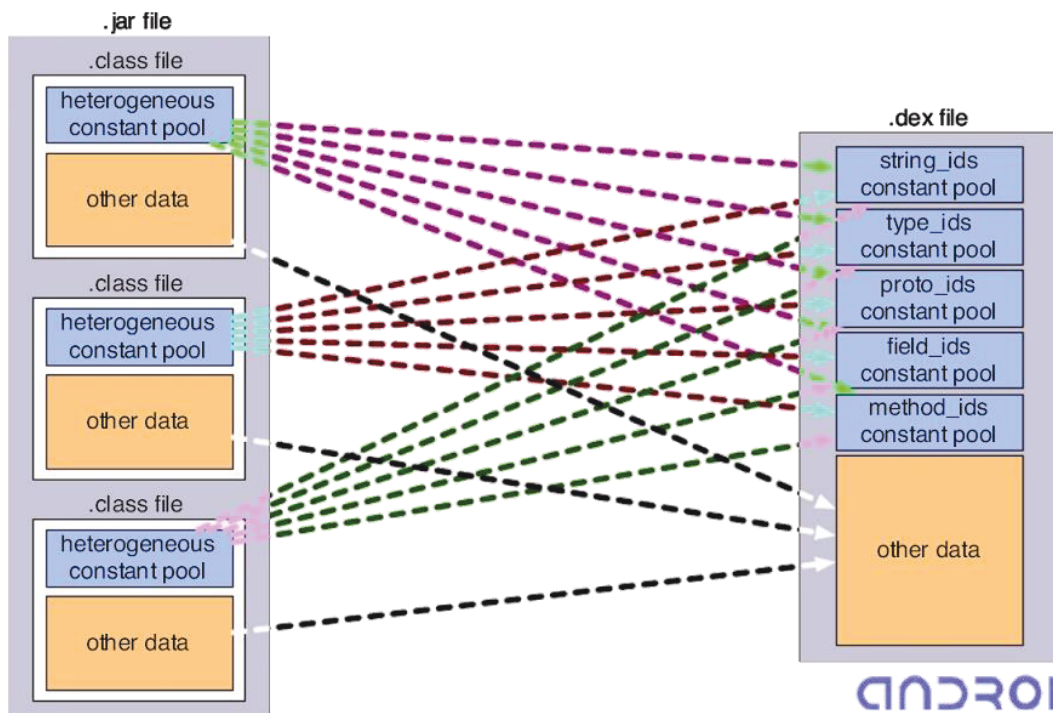


Ilustración 42
Detalle Fichero
DEX

Actualmente Dalvik tiene sus días contados pues Google en sus versión 4.4-Kitkat presentó una nueva maquina Virtual, **ART** ⁶⁰, La principal diferencia de ART con respecto a Dalvik el sistema de compilado, ART usa AOT y Dalvik utiliza JIT. Detallamos a continuación cada uno de los sistemas.

ART vs Dalvik / AOT vs JIT

Dalvik- JIT	ART- AOT
<p>JIT- “just in time compilation”</p> <p>Cada vez que una app es ejecutada, la parte del código que se necesita para su ejecución es compilada a código máquina, según se vaya usando la app, se irá incrementando el código a utilizar generando código adicional según sea requerido.</p> <p>Esta técnica tiene como puntos a favor el bajo uso de recursos, menos uso de memoria y disco.</p>	<p>AOT, cuando una aplicación es instalada esta se transforma de bytecode a un “binario dependiente”, un precompilado que se hará al instalar una aplicación. Esto hará aumentar el tiempo de instalación, pero hará desaparecer el “lag” de apertura de las aplicaciones, a su vez, este precompilado en código máquina, optimiza el uso de la CPU con lo que a costa de un mayor tiempo de instalación y un consumo mayor memoria , reducimos drásticamente el tiempo de carga de las apps y reducimos el consumo de batería.</p>

Ilustración 43Tabla Comparativa DALVIK- ART

Podemos ver como el uso de la Maquina Virtual ART, al ser menos dependiente de procesador y tener una pre compilación optimizada mejora en gran medida el rendimiento.

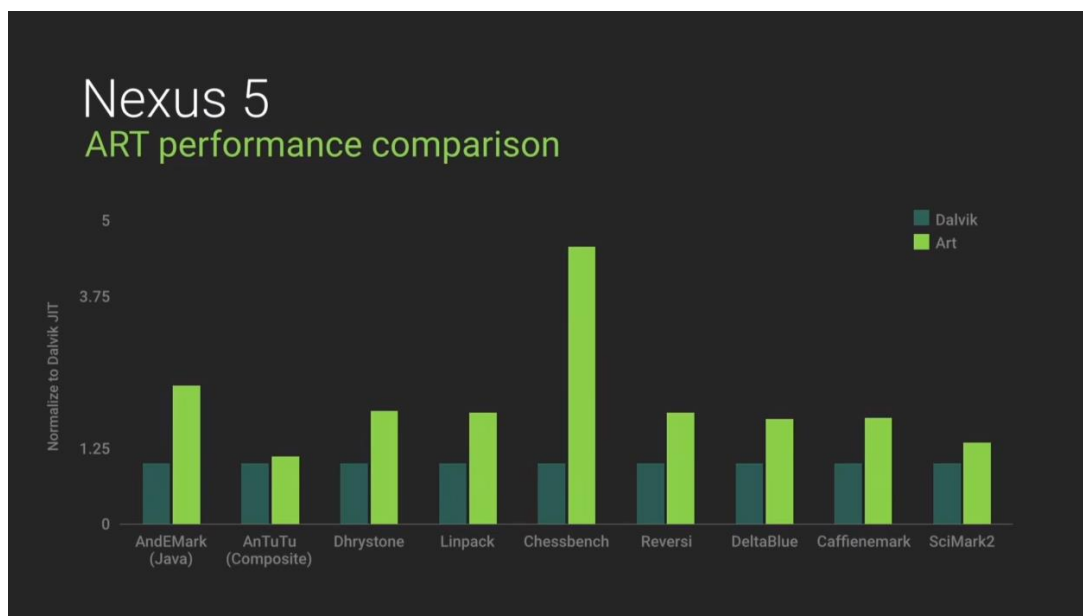


Ilustración 44 Comparativa rendimiento Dalvik- ART

3.1.3. Librerías Nativas.

Se corresponde con las librerías utilizadas en Android. Están escritas en C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo constituyen el corazón de Android. Entre ellas podemos destacar :

- **System C library:** una derivación de la librería BSD de C estándar, adaptada para dispositivos embebidos .
- **Media Framework:** aporta los códec de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** navegador Web embebido en Android y en la vista Webview.
- **SGL:** motor de gráficos 2D.
- **Librerías 3D:** basada en OpenGL ES 1.0 API.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite:** motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** servicios de encriptación *Secure Socket Layer*.

3.1.4. Framework de Aplicaciones.

Representa el conjunto de herramientas de desarrollo. Toda aplicación utilizan este conjunto de API framework que presta este nivel. Esta capa esta diseñada para la reutilización máxima de los componentes, las aplicaciones pueden utilizar las capacidades ofrecidas por la capa o sobrescribirlas para adaptarlas a los requerimientos. Entre los servicios más importantes podemos destacar:

- **Views:** parte visual de los componentes.
- **Resource Manager:** Acceso a recursos que no son código.
- **Activity Manager:** manejo del ciclo de vida de la aplicación.
- **Notification Center:** permite a las apps mostrar alertas en la barra de estado.
- **Content Provider:** mecanismo de intercambio de datos, contenedores de datos.

3.1.5. Aplicaciones.

Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo. Todas pueden utilizar servicios, API y librerías de niveles anteriores.

El desarrollo de la aplicación tiene como fundamentos:

- Android es un sistema operativo multi-usuario, en el que cada aplicación tiene su propio usuario.
- Por defecto, cada aplicación tiene su propio userID, el cual es desconocido por la aplicación (el ID solo lo usa el SSO). El sistema asigna permisos a todos los ficheros de la aplicación para que solo ese user ID pueda acceder a ellos.
- Cada proceso tiene su propia máquina virtual, así que las aplicaciones se ejecutan aisladas unas de otras.
- Por defecto, cada aplicación se ejecuta en su propio proceso de Linux. El proceso se crea según lo necesita la aplicación y se destruye si otras aplicaciones necesitan los recursos.

3.2. El patrón MVC.

En Android se sigue el patrón de Arquitectura “Modelo – Vista-Controlador”, este define la organización separada e independiente del Modelo (Objetos y lógica de Negocio), la Vista (interfaz de usuario) y el Controlador (workflow de la aplicación). Esto divide el sistema en tres capas haciendo posible la encapsulación de los datos, la interfaz y la lógica de negocio de la aplicación.

MODELO	VISTA	CONTROLADOR
Contiene el núcleo y funcionalidad de la aplicación, es independiente de las vistas y los controladores, establece la funcionalidad para la que se crea la App. Establece la capa más baja de datos encapsulando el estado de la aplicación.	Es la presentación del Modelo. Interactúa con el modelo mediante el controlador, puede ser notificada ante la presencia de cambios en el modelo para que se repinte. Generalmente suele estar generada utilizando los XML, aunque también puede hacerse de manera programática.	La vista avisa a el controlador de las solicitudes del usuario, este solicita al modelo los cambios y una vez que el modelo los tiene el controlador informa a la vista para que se repinte.

Ilustración 45 Tabla detalle Modelo MVC

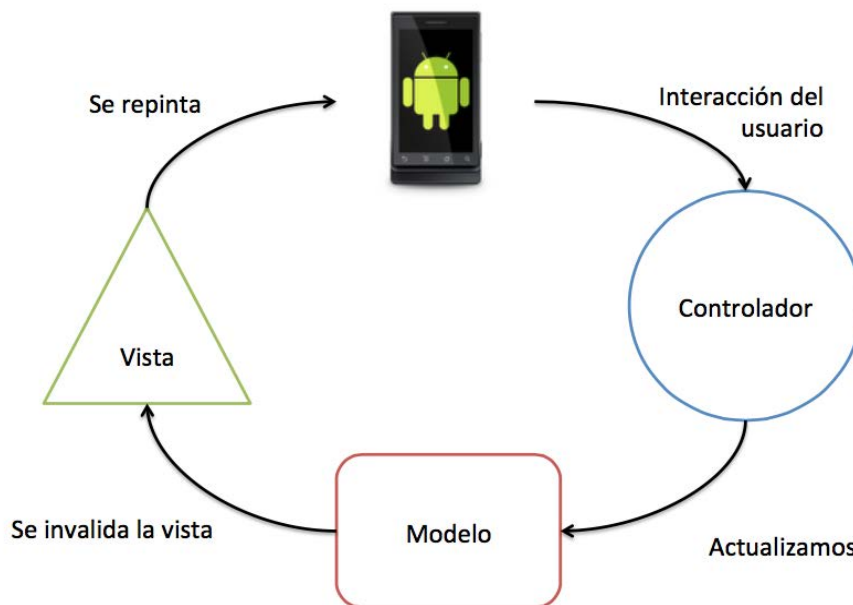


Ilustración 46 Detalle MVC

La interfaz de usuario recibe funcionalidad gracias a los eventos “event Listeners”, que son interfaces definidas de la clase View que contienen un método callback, que se llamara cuando el usuario interactúe con él.

3.3. Elementos que componen una aplicación Android.

Una aplicación Android esta compuesta por una serie de elementos “débilmente acoplados”. Todos ellos están relacionados en el archivo “manifests”, en donde se especifican los requisitos hardware (si necesita cámara, GPS...), librerías externas y permisos que necesita la app.

3.3.1. Activity

Esta es la parte que implementa la lógica de la aplicación y enlaza la capa de presentación. Las Activitys utilizan FRAGMENTS y VIEWS para mostrar la información y responder a las interacciones del usuario.

Android maneja las activitys como una Pila, cada nueva actividad se superpondrá a la anterior, como Android es multitarea y un usuario puede cambiar de actividad, a la vez que una misma aplicación puede tener declaradas varias Activitys se necesita un mecanismo de control dependiente del estado en que se encuentre la actividad. En base a esto, toda actividad posee un **ciclo de vida** en donde se disparan los diferentes estados según corresponda.

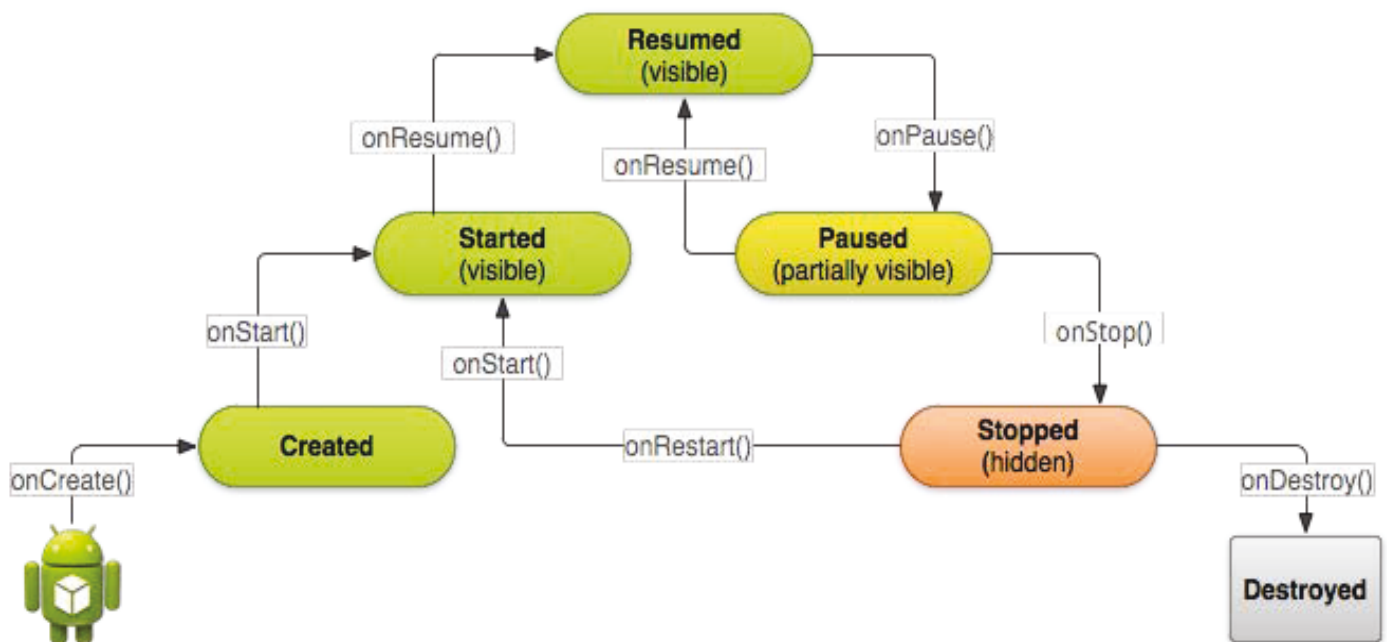


Ilustración 47 Detalle Activity

3.3.2. Servicios

Son un componente sin interfaz de usuario, es decir, sin Views ni fragments asociados. Ejecutan la mayoría de las acciones que la aplicación necesita realizar en Background. Pueden estar activos incluso si las actividades de la aplicación no lo están.

Los servicios son muy parecidos a una actividad pero sin la interfaz visual, con lo que tiene un ciclo de vida sin estas características.

Un servicio puede estar en 2 estados, como se observa en la figura.

Started: el servicio está iniciado y se ejecutará de manera indefinida hasta que termine su cometido, en ese momento procederá a pararse.

Bound: Diferentes componentes se pueden enlazar al servicio, estará siempre en ejecución mientras tenga algún componente enlazado, cuando todos los componentes asociados hayan llamado a UnbindService, el servicio se destruye automáticamente.

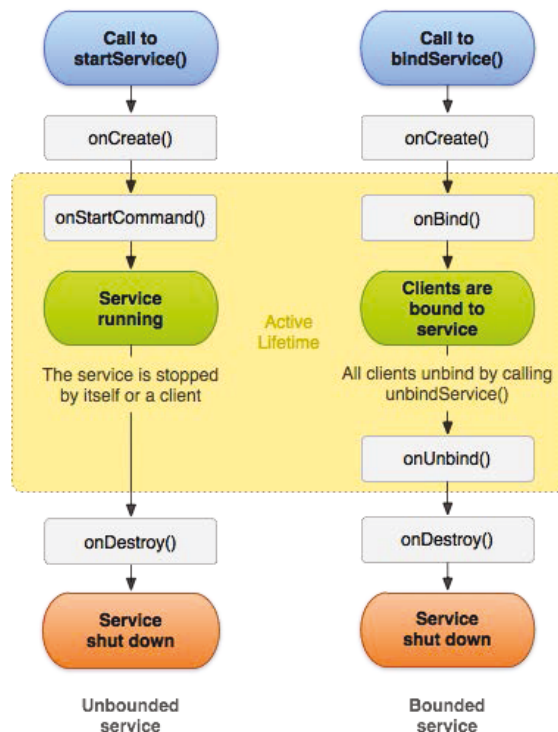


Ilustración 48 Ciclo Vida Servicio

3.3.3. Content Providers

Este elemento permite construir almacenamiento de datos persistente y modular; se pueden utilizar en distintos puntos de la aplicación. Normalmente interactúan con Bases de Datos SQLite o con llamadas remotas. Se pueden configurar diferentes content providers para que otras aplicaciones puedan acceder a ellos, de forma análoga a como nuestra aplicación accede a los facilitados por otras aplicaciones.

Existen content providers ofrecidos por el sistema operativo, de todos ellos los más comunes son el de contactos, el cual contiene toda la información referente a la Agenda de nuestro teléfono o la Agenda con el calendario y eventos personalizados.

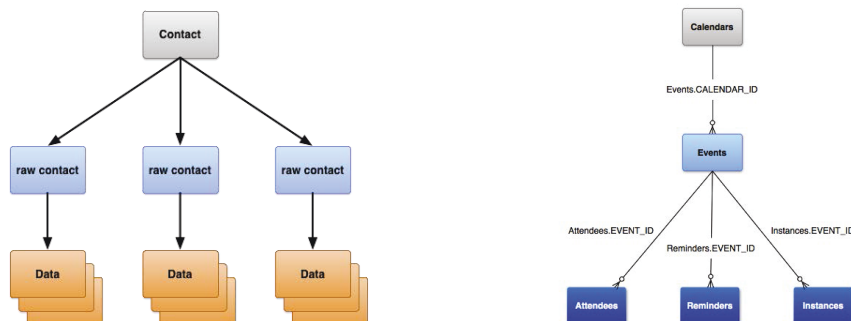


Ilustración 49 y 50
Diseño Content
Providers

3.3.4. Intents

Son un Framework, estructura de datos utilizada para pasar datos entre actividades, servicios o aplicaciones. Se usan Intents para iniciar o parar actividades o servicios, enviar mensajes en los broadcast o pedir la ejecución de una acción en espera de un resultado.

Los Intents pueden ser implícitos o explícitos, la diferencia es que en los explícitos se especifica el nombre del componente llamado en vez de usar un “Intent filter” para encontrar el componente llamado. Los Intents Pueden contener:

- ✓ **COMPONENT NAME:** nombre completo de la clase y componente declarados previamente en el manifest.
- ✓ **ACTION:** acción a realizar, especificada mediante constantes, ACTION_EDIT, ACTION_SEND, ACTION_VIEW
- ✓ **DATA:** se trata de una *Uri* , por ejemplo para lanzar una llamada se define ACTION_CALL y en este campo el teléfono a llamar.
- ✓ **CATEGORY:** información adicional sobre el tipo de componente.
- ✓ **EXTRAS:** es un mapa de pares clave-valor.
- ✓ **FLAGS:** existen multitud de flags aplicables según sea la utilidad del Intent, por ejemplo si estamos lanzando una actividad que no queremos que se quede en la pila de actividades, podemos marcar FLAG_NO_HISTORY.

3.3.5. Broadcast Receivers

Son elementos que están escuchando Intents, que cumplen los criterios que se especifican. Así pueden iniciar la aplicación en reacción a cualquier Intent que reciben.

Existen dos tipos de llamadas a los BroadcastReceivers, las que programamos nosotros mismos y que están dentro de la aplicación y las que se reciben desde el Sistema operativo, como por ejemplo podría ser, si hiciéramos una App despertador , programaríamos para recibir el broadcast de “teléfono en silencio” para poder advertir al usuario.

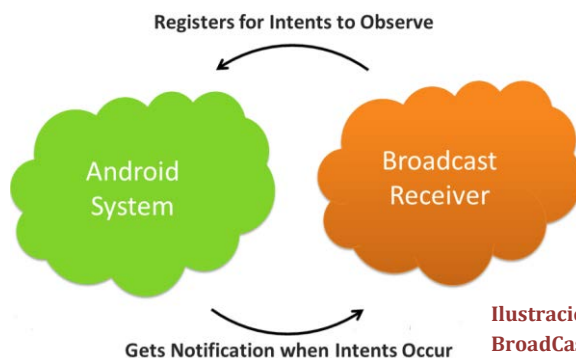


Ilustración 51
BroadCast
Receiver

3.3.6. Widgets

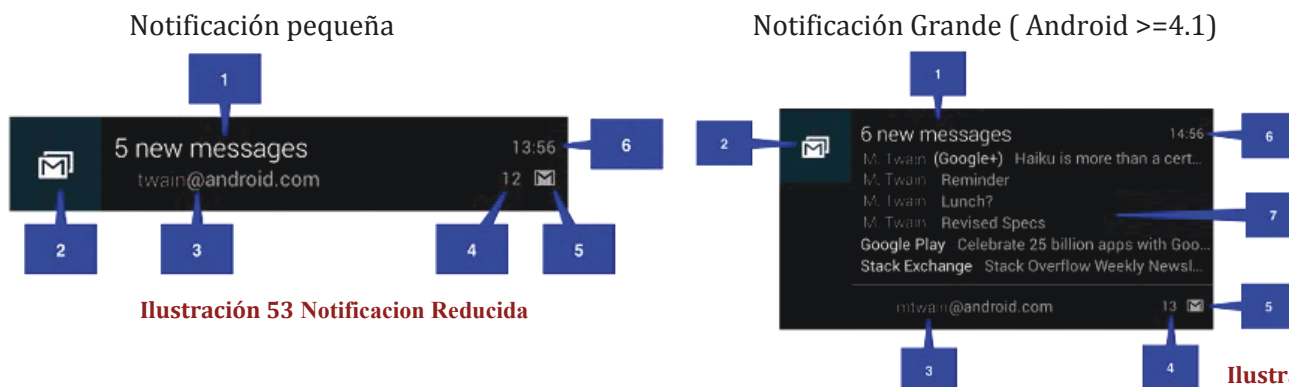
Son aplicaciones que se pueden añadir a la pantalla principal del dispositivo. Generalmente suelen tener animaciones y una especial atención en el aspecto gráfico. Añaden funcionalidad a el home de Android gracias a que en ellos podemos mostrar cualquier dato una clara ventaja de Android Frente a iOS, la fácil creación y adaptación de entornos dinámicos y personalizados sobre las vistas del launcher .



Ilustración 52 ejemplo widget

3.3.7. Notificaciones

Permiten avisar al usuario sin interrumpir la actividad que están ejecutando en ese momento. Es la técnica preferida para llamar la atención del usuario cuando nuestra aplicación no es visible. Las notificaciones suelen ser lanzadas en Services o Broadcast Receivers, que son componentes que no tienen interfaz de usuario.



1- Título
4- Información

2- Icono grande
5- Icono pequeño

3- Contenido
6- Fecha de la notificación

Ilustración 54
Notificación
Completa

3.3.8. El archivo Manifest.xml

Cualquier proyecto Android necesita incluir el fichero AndroidManifest.xml, en él se incluyen todas las declaraciones de los componentes vistos hasta ahora en este apartado, si alguno no esta definido, esto generará un error en la ejecución. En el se incluye toda la Meta información referente a la aplicación, el icono de la

aplicación, el numero de versión, los permisos necesarios que deberán ser aceptados por el usuario, hardware necesario, resoluciones soportada etc.

3.3.9. Recursos estáticos y localización

En todo proyecto, es una buena practica mantener los recursos estáticos, textos, fotos, etc.... separados del código fuente. Android soporta la externalización de estos recursos mediante una localización específica de los diferentes elementos.

En la aplicación Hispania uc3m se puede observar como en eclipse existen estas carpetas y como dentro de ellas se encuentran agrupados los recursos según su tipo.

3.3.9.1. res/values

En esta carpeta se guardan los valores más básicos de la aplicación

Valores Simples

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">To Do List</string>

    <plurals name="androidPlural">
        <item quantity="one">One android</item>
        <item quantity="other">%d androids</item>
    </plurals>

    <color name="app_background">#FF0000FF</color>

    <dimen name="default_border">5px</dimen>

    <string-array name="string_array">
        <item>Item 1</item>
        <item>Item 2</item>
        <item>Item 3</item>
    </string-array>

    <array name="integer_array">
        <item>3</item>
        <item>2</item>
        <item>1</item>
    </array>
</resources>
```

Estilos y temas

```
3.3.9.1. <?xml version="1.0" encoding="utf-8"?>
<resources>

    <style name="AppTheme" parent="android:Theme.Light" />

    <style name="base_text">
        <item name="android:textSize">14sp</item>
        <item name="android:textColor">#111</item>
    </style>

    <style name="small_text" parent="base_text">
        <item name="android:textSize">8sp</item>
    </style>

</resources>
```

res/drawables

En esta carpeta se almacenarán todos los recursos visuales de la aplicación ya sean Bitmaps o ficheros XML que definen un recurso visual, una tabla, un recuadro, un borde...

Cabe destacar que este recurso se le pueden añadir terminaciones como puede ser res/drawables-large o res/drawables-hdpi que especifican diferencias en la pantalla, por resoluciones o por formato. Esto es así pues pueden ser necesarios distintos recursos visuales para amoldarse a todos los formatos disponibles.

3.3.9.3. res/layout

Aquí se guardaran todos los XML referentes a el contenido visual de nuestra aplicación. En ellos se definirá mediante el uso de diferentes elementos las vistas de cada una de las pantallas.

Android facilita una serie de elementos contenedores como son los “relative, horizontal o vertical layout” que sirven para establecer un orden entro otros objetos más tangibles como pueden ser botones, campos de texto, listados, imágenes...

```
<?xml version="1.0" encoding="utf-8"?>
<.ViewGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+[package:]id/resource_name"
    android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
    android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
    [ViewGroup-specific attributes] >
    <View
        android:id="@+[package:]id/resource_name"
        android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
        android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
        [View-specific attributes] >
        <requestFocus/>
    </View>
</ViewGroup>
<View />
</ViewGroup>
<include layout="@layout/layout_resource"/>
</ViewGroup>
```

3.3.9.4. raw

Se pueden guardar ficheros que podremos leer directamente como ficheros binarios y streams.

3.3.9.5. Assets

Parecido al anterior, pero permite una estructura de directorios. Se utiliza comúnmente para guardar los ficheros de WebApps.

3.3.10. Otros Recursos y localizaciones.

A parte de los recursos estáticos como imágenes menús, o elementos visuales existen otras ubicaciones que son de gran relevancia a la hora de realizar un proyecto.

3.3.10.1. **src**

Esta será la localización de todas las clases del proyecto, es recomendable establecer algún tipo de orden que organice el espacio de trabajo, una buena practica puede ser la agrupación por unidades funcionales, o por módulos de la aplicación.

3.3.10.2. **libs**

En esta localización se encontraran todas las librerías de terceros que sean necesarias para la ejecución de nuestra aplicación. El compilador tomará los .jar aquí existentes y los integrará con el proyecto.

3.3.10.3. **gen**

Esta ubicación será autogenerada por el compilador, en ella se crearán los archivos R.java necesarios para el proyecto. Esta carpeta nunca debe ser tocada por el programador pues los cambios que realice serán borrados en una siguiente compilación del proyecto.

3.4. Fragmentos

Android 3.0 se desarrolló como una orientación específica hacia las Tablets. Estos dispositivos tienen unas pantallas mucho más grandes que los teléfonos móviles, por lo que presentan un gran abanico de posibilidades en cuanto al juego que permiten dar con las vistas.

Hasta Android 3.0 toda aplicación de Android estaba realizada con Activitys, pero estas, por su diseño nos permiten un diseño muy limitado en cuando a las pantallas, esto es debido a que cada Actividad posee únicamente un elemento “vista”, preservando así el Modelo Vista Controlador previamente explicado. Como solución a esta situación de una vista única por actividad, se desarrollaron los Fragmentos, los cuales representan una porción de la Interfaz de usuario de la actividad. Se podría decir que los Fragmentos, son una forma modular de construir la interfaz de usuario, una especie de “sub-Activity” que puede ser reutilizable desde varias Actividades. Es por ello que un Fragmento siempre esta contenido en una actividad y su ciclo de vida depende de la actividad padre.

La gran utilidad de los fragmentos es que nos permiten construir vistas flexibles, escogiendo los elementos que sean oportunos para cada pantalla, pudiendo combinar fragmentos con activitys, o como se lleva por tendencia cada vez más, hacer activitys más grandes con multitud de fragmentos que pueden ser integrados en ellas. Por poner un claro ejemplo de la utilidad de los fragmentos, cualquier aplicación de correo visto en Tablet, que cuando esta en horizontal marca en un lado el listado de correos y en el otro el detalle, esta realizado mediante fragmentos.

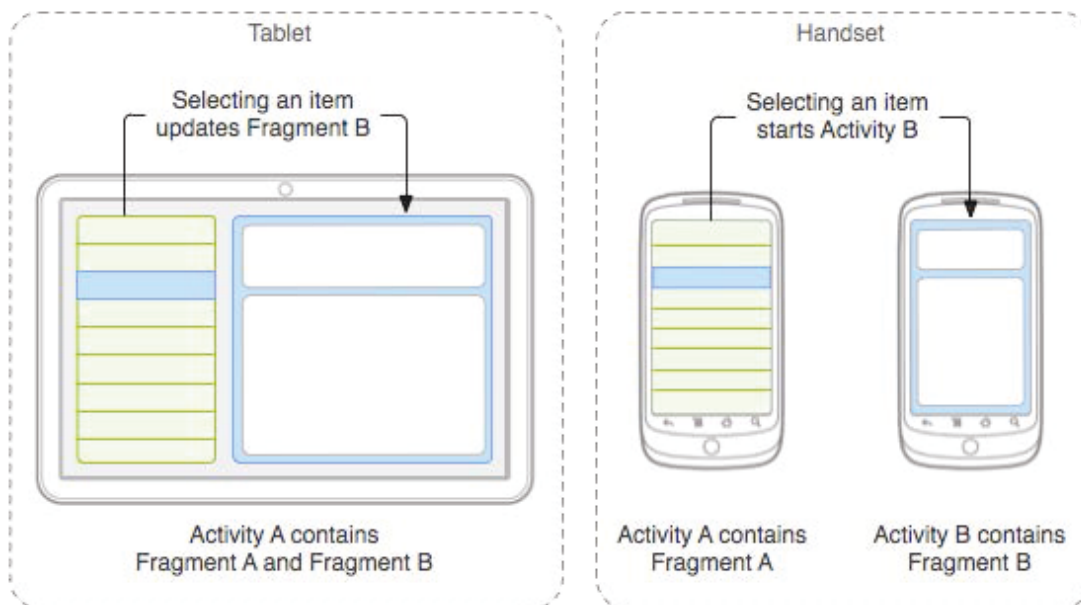


Ilustración 55 Detalle Fragmentos

Como Hemos comentado previamente los fragmentos van integrados en el ciclo de vida de la Actividad padre, y hemos dicho que son partes modulares de esta, con lo que poseen, un ciclo de vida que se ejecuta dentro del ciclo de la actividad.

1. **On Create():** en este punto se deben inicializar los componentes esenciales.
2. **On Create View():** Es llamado cuando el fragmento tiene que crear su interfaz de usuario.
3. **On Pause():** El fragmento esta siendo detenido, bien sea porque la actividad padre va a entrar en onpause también o solamente va a ser el fragmento pues va a ser reemplazado por otro (por ejemplo). En este momento deberemos guardar cualquier información que queramos añadir de forma persistente al modelo.
4. **OnSaveInstanceState():** Aunque no pertenece al ciclo de vida propiamente dicho, es importante mentarlo pues este método nos guarda el estado del fragmento, así si por ejemplo cambiamos la orientación del dispositivo, el fragmento se destruirá y volverá a ser generado pero en su método onCreate u onCreateView podremos recuperar el estado.

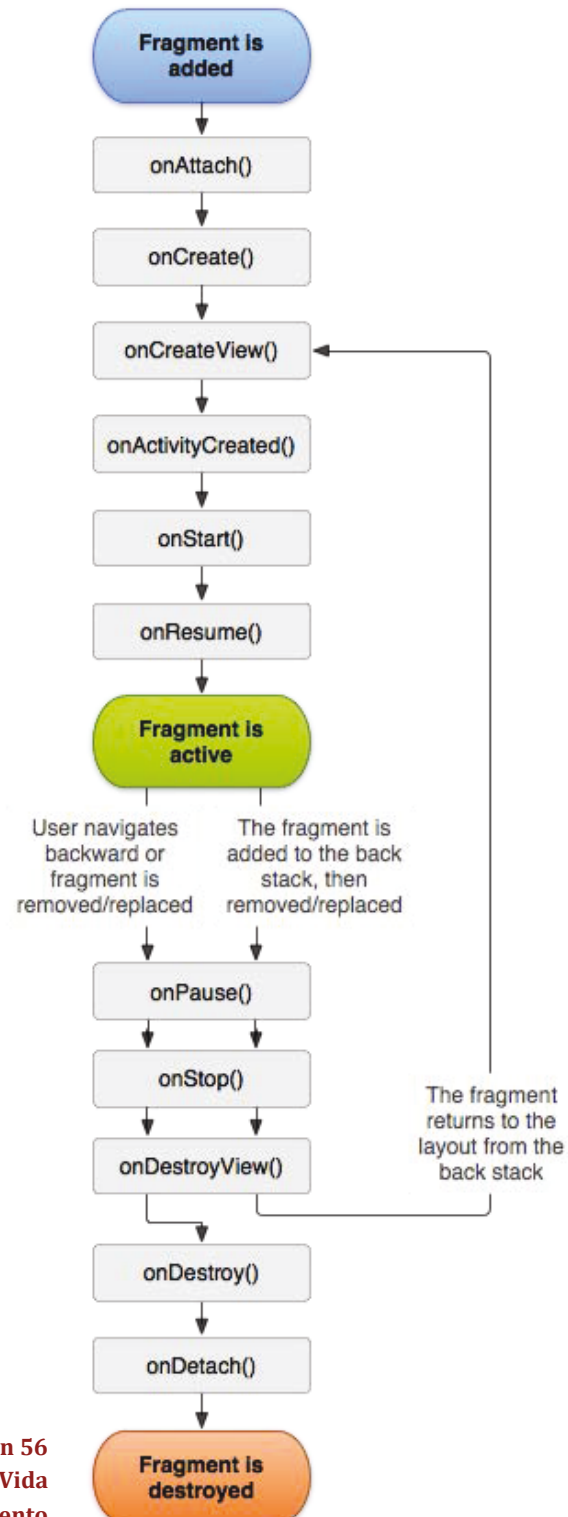


Ilustración 56
Ciclo de Vida
de Fragmento

3.5. Threads y Async-Task

Debido a que la aplicación Hispania uc3m hace un uso constante de hilos y ejecuciones en segundo plano, debido a su carácter necesidad constante de comunicación con el backend vamos a ver las diferentes técnicas que nos ofrece Android para realizar las Tareas en Background. A su vez, creo que este es un punto de vital importancia, pues viendo como evolucionan las capacidades hardware, hacia cada vez más núcleos, es importante conocer y entender los diferentes mecanismos que nos van a permitir hacer uso de estas arquitecturas. En Android, por defecto, toda ejecución se hará en el hilo principal, lo cual nos puede generar inestabilidades o bloqueos de la interfaz en caso de realizar procesos pesados o usos de red. Esta situación hace que el sistema operativo pueda entender como que el hilo esta bloqueado nos alertará como si el sistema hubiese dejado de responder.

THREADS: Este método es heredado de Java, utiliza el método run de una clase que implemente la interfaz runnable, aunque esta muy extendida su utilización no ofrece ninguna optimización sobre Dalvik/Art al tratarse de simple herencia de JAVA. Para actualizar la interfaz de usuario hay que llamar a “runOnUiThread” para que el sistema operativo pase la ejecución al hilo principal. Depende el uso que se quiera, puede ser útil su uso, por ejemplo en sencillas llamadas a la red o tareas livianas, pues el código que genera es más sencillo y manejable.

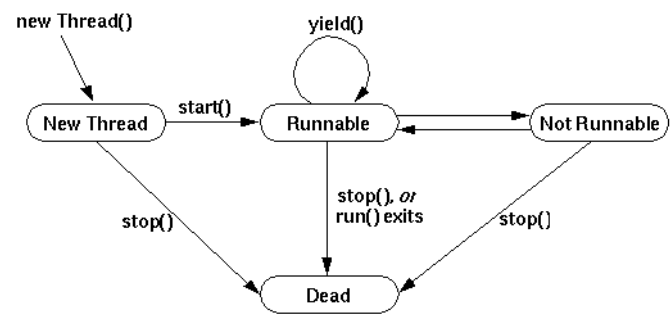


Ilustración 57 Detalle Thread

ASYNC TASK : Esa una clase abstracta propia de la biblioteca de Android que nos permite una organizada ejecución en 4 pasos:

1. *onPreExecute*: se ejecuta en el hilo principal, se usa para hacer los cambios necesarios en la UI antes de la ejecución
2. *doInBackground*: se invoca en un hilo separado, en este paso se realizaran todas las tareas pesadas o de red que necesitemos sin que interfieran con el hilo principal o la interfaz. Se puede opcionalmente llamar a *publishProgress*
3. *onProgressUpdate*: si el método de Background llama a *publishProgress*, qué se ejecutará en el hilo principal, es usado por si queremos informar en la interfaz de usuario sobre el progreso de Background.
4. *onPostExecute*: una vez terminado el trabajo del hilo secundario, se pasa al principal con el resultado.



Ilustración 58 Detalle Async Task

3.6. El Cliente HTTP

En Android podemos utilizar dos clientes de red:

- 1) HttpURLConnection: recomendado en Android 4.0 en adelante.
- 2) Apache HttpClient: Utilizado en las versiones más antiguas, aunque actualmente es plenamente compatible.

Ambos clientes soportan de forma nativa http con SSL, streaming, timeouts configurables por el programador, IPv6 y connection pooling.

Android, por política de permisos y no permitir una conectividad de una aplicación no deseada, por lo que en caso de querer que nuestra aplicación tenga acceso a internet deberemos declarar explícitamente en nuestro archivo manifest el permiso. Además, como para cualquier tipo de conectividad siempre es recomendable antes saber si esta existe, tendremos que asegurarnos también el poder obtener el estado de la red.

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Las conexiones http se realizarán siempre en un thread aparte, si intentásemos realizarlas en el hilo principal, a no ser que lo especifiquemos en el manifest y forcemos esta mala práctica, recibiremos un error en tiempo de ejecución. Una vez que la conexión haya sido terminada habremos recibido un inputStream, el cual deberemos recuperar según el dato que estemos esperando(imágenes, Strings, un XML, o Json...).

ejemplo para
recuperar una
imagen:



```
InputStream is = null;  
...  
Bitmap bitmap = BitmapFactory.decodeStream(is);  
ImageView imageView = (ImageView) findViewById(R.id.image_view);  
imageView.setImageBitmap(bitmap);
```

```
// Reads an InputStream and converts it to a String.  
public String readIt(InputStream stream, int len)  
    throws IOException, UnsupportedEncodingException {  
    Reader reader = null;  
    reader = new InputStreamReader(stream, "UTF-8");  
    char[] buffer = new char[len];  
    reader.read(buffer);  
    return new String(buffer);  
}
```

ejemplo
para
recuperar
una
cadena de
texto tipo
String.



CAPÍTULO 4

HERRAMIENTAS DE DESARROLLO

4. HERRAMIENTAS DE DESARROLLO

Para la realización de este proyecto hemos tenido que utilizar diversas herramientas para cubrir los diferentes campos de diseño, programación y backend.

Aunando las características de cada una obtenemos nuestro puzzle específico sobre el que se genera todo el *ingenio* necesario, una base sólida de desarrollo que hace plausible la consecución de los diferentes objetivos sobre los que se apoya este proyecto.

4.1. Herramientas de diseño

Dentro de estas, se establecen el conjunto de herramientas que han sido necesarias para dar vida al diseño de la Aplicación, con ellas hemos creado desde el boceto inicial, gracias al cual hemos estudiado el sistema de Juego, hasta la elección de colores y estilos aplicados en las diferentes pantallas.

4.1.1. Lápiz y papel.

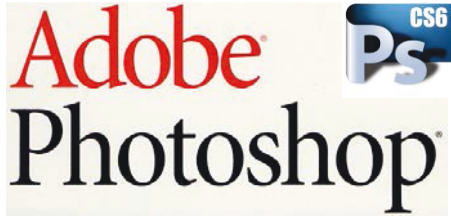
Aunque no son una herramienta informática ni tecnológicamente avanzada, creo que es fundamental mentarla. A día de hoy uno de los grandes fallos que existen en el mundo de la informática, sobre todo en proyectos de reducido tamaño, es el arranque de proyectos con potentes herramientas que gracias a un aspecto visual avanzado nos ayudan a generar unos primeros pasos más completos y que dan una sensación de un trabajo más elaborado, no obstante, esto es una mera apreciación pues estaremos en una fase inicial del proyecto atribuyendo unos valores que no son los que se corresponden con una etapa temprana de desarrollo.

La utilización de un primer boceto en papel nos permite:

- ✓ Realizar una primera aproximación del esqueleto de la aplicación
- ✓ Idear una navegación de pantallas intuitiva y jerarquizada.
- ✓ Permite múltiples Bocetos, Gestionar muchas ideas en un espacio de tiempo reducido.
- ✓ Rápida corrección sobre los bocetos.
- ✓ Establecer una atención sobre la idea Base del proyecto, Jugabilidad y navegación. Se dejan fuera aspectos del diseño muy visuales pero no relevantes, colores, formas...

Una vez que tenemos un primer diseño sobre papel, ya si podremos empezar a utilizar herramientas informáticas y centrarnos en características más específicas del diseño.

4.1.2. Adobe Photoshop.



Como primera herramienta de diseño se ha optado por utilizar uno de los estándares del mercado el “Adobe Photoshop”, más concretamente se ha utilizado su versión CS6.

A primera vista esta herramienta nos queda muy potente para el cometido de la aplicación pues su principal target es el mundo del retoque fotográfico, no obstante, su versatilidad la hacen indispensable para generar mockups y poder realizar distintas configuraciones de formas y colores que nos ayudan a ver in situ un diseño final del prototipo. A su vez sus herramientas de selección, recorte, giro etc.... nos ayudan a realizar de una manera profesional y exacta cambios sobre la iconografía, sombras etc....

Para realizar la iconografía se ha optado por la búsqueda de series de iconos gratuitos existentes en internet, no obstante se han aplicado ciertas mejoras como son la adecuación de los colores o la aplicación de sombras.



El icono de la papelera era necesario transformarlo a varios colores para adecuarse a cada diseño de pantalla. A su vez se le aplico efecto de sombra para mejorar su acabado.



A el icono de ranking se le ajustó el color, se le agregó sombra y se realizó un suavizado de las aristas.

Este icono, utilizado para detallar la pista para escribir la palabra, se creó a partir de un icono de opciones y un lapicero, aplicándose sus correspondientes efectos de sombra y suavizado.



Icono de partida ganada. Se optó por una simbología tomada a partir de una imagen de una corona de “julio Cesar” a la que se le agregó una escarapela con el logo de la universidad.



Los iconos de multijugador se crearon mediante la duplicidad y agregado del icono de un jugador, cambiando su paleta colorífica.



Ilustración 59 Tabla Ilustraciones y Cambios

4.2. Herramientas de Programación

Pasamos a detallar las diferentes herramientas utilizadas para generar el código de la aplicación, gracias a ellas se ha podido programar la lógica de la aplicación.

Para la realización de este proyecto podemos optar por el uso varias plataformas de desarrollo que encontramos en el mercado, entre ellas podemos destacar Eclipse y Netbeans como referentes en el mercado por su compatibilidad con diversos lenguajes, nivel de madurez y apoyo recibido por la comunidad., no obstante de todas ellas solamente podremos utilizar las que sean compatibles con el Android SDK, ya sea de manera nativa o mediante plugins.

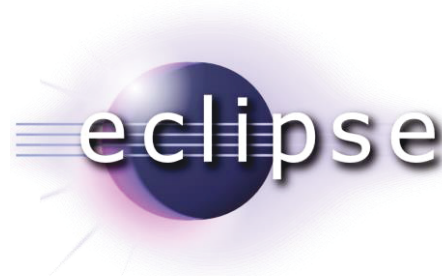


Ilustración 60 Ilustración filtrado de Ides

No obstante para el entorno que nos acata, encontramos bastantes diferencias entre ellas, empezando desde la compatibilidad con el plugin de ADT el cual, en sus inicios se desarrollo para eclipse, aunque puede ser instalado en otros. Android Studio, actualmente es referente y va a ser la plataforma futura de desarrollo gracias a numerosas mejoras que presenta contra eclipse, no obstante se encuentra en un estado de relativa inmadurez y muchas de las ventajas que presenta como puede ser integración nativa con Gradle , que facilita el tedioso mundo de los imports, presentan, según la comunidad todavía pequeños problemas e incompatibilidades que reducen la estabilidad del proyecto.

No pudiendo beneficiarme completamente de estas ventajas, pues entre otros, la librería utilizada por App42 no es compatible con Gradle, he optado por el entorno Eclipse, el más maduro actualmente para desarrollo Android que podemos encontrar y donde la documentación específica es clara y detallada a la vez de ser más conocido para el programador.

4.2.1. Eclipse



Eclipse esta conceptualizado como una plataforma base para el desarrollo en cualquier lenguaje, es una plataforma de integración de herramientas de desarrollo proporcionando elementos de gestión de espacios de trabajo, y herramientas que permiten la ejecución, depuración y despliegue de aplicaciones.

El crecimiento sobre esta base se da mediante plugins que aportan las funcionalidades específicas de cada lenguaje.

En el caso de JAVA, lenguaje utilizado en este proyecto se ha utilizado el plugin específico proporcionado por Google para la realización de aplicaciones Android en lenguaje Java, el denominado ADT.

Las principales características que han ayudado a eclipse a convertirse en un referente entorno de las aplicaciones software de entorno de desarrollo son⁵⁷:

- **Clara orientación al Proyecto:** Su interfaz esta enfocada al proyecto, cada proyecto tiene claramente definido el conjunto de recursos que lo integran, el código fuente, la configuración, árbol de directorios...

- **Basado en perspectivas específicas:** cada proyecto tiene una pre configuración de ventanas interrelacionadas entre sí que en conjunto a sus editores generan un entorno de trabajo óptimo.
- **Depurador de código integrado:** Eclipse, posee por defecto un potente e intuitivo depurador, cuenta con perspectivas específicas para este cometido y permite un control casi absoluto de los saltos de código.
- **Extensa Colección de plugins:** Se ofrecen multitud de plugins que facilitan tareas en eclipse, facilitados tanto por los propios desarrolladores del IDE como por terceros.

Desde un punto de vista específico de Android, la nativa compatibilidad con el ADT de Google para el desarrollo de Android y su completo suite de herramientas denominado “Android Tools” ofrecen un entorno óptimo para el desarrollo JAVA orientado a el desarrollo Android debido a:

- **Coloreado del código:** el reconocimiento sintáctico y coloreado facilita la visibilidad y legibilidad del código.
- **Formateo automático:** Aunque esta característica es heredada del plugin JAVA JDT- PLUGIN nos permite formatear el código de una manera que su legibilidad se mejora enormemente.
- **La Ayuda y tips sobre sentencias** son constantes y actualizadas, basta con mantener Android tools actualizado para saber que métodos están o van a estar deprecados, avisos sobre que funciones están disponibles a partir de diferentes Api-level o recomendaciones de mejora de compatibilidad en los layouts, entre otros, facilitan al programador la creación del código. Esa circunstancia genera de manera intrínseca un código de más calidad.

4.2.2. Herramientas de tratamiento de Datos.

Para generar la base de datos, necesitamos un formato que sea entendible por nuestro proveedor back-end. En un principio nuestro servidor de preguntas sobre el juego, va a estar alojado en Parse.com y supuestamente admite como formatos de entrada CSV y J-son para crear las tablas de una manera sencilla.

Nuestra tabla de datos, con las preguntas que se realizarán en el Juego se ha realizado con la conocida herramienta Microsoft Excel, donde su organización tipo tabla y la fácil aplicación de filtros sobre las columnas la hacen idónea para saber cuantas preguntas van por nivel y así mantener una calidad aceptable sobre nuestra base de datos.

En la practica ha sido imposible que la consola de Parse.com nos realizara una importación correcta de los datos en formato CSV, con lo que se tuvo que establecer un método alternativo que nos permitirá utilizar una subida de grandes tablas. Esta solución vino de la mano de otro de los formatos que admite como entrada, JSON, con lo que para poder utilizar la subida por lote facilitada en Parse.com hubo que hacer una transformación de los datos de CSV a JSON, para lo cual se utilizó la web gratuita de : <http://www.convertcsv.com/csv-to-json.htm> .

Una vez que teníamos nuestro fichero Json se presento un nuevo problema con los el fichero Json obtenido, la codificación generaba caracteres erróneos en la importación.

Se tuvo que utilizar el programa de edición de texto Ultra-edit en su versión gratuita que ofrece un conversor de ANSI a utf-8. Finalmente este archivo JSON sí que fue aceptado por las herramientas de Gestión de Parse.com y permitió la incorporación masiva de datos a las tablas de backend.

CAPÍTULO 5

DESARROLLO HISPANIA UC3M

5. HISPANIA UC3M, ANALISIS Y DISEÑO E IMPLEMENTACIÓN

En este capítulo se detallarán todos los pasos y etapas que conforman el proceso de desarrollo de la aplicación. Análisis, diseño e implementación.

También entraremos en profundidad en todos los aspectos relevantes de la App, como se ha organizado el proyecto, que entorno de desarrollo se ha utilizado, los requisitos para un correcto desarrollo y ejecución de la app así como de todo lo que se considere relevante para su correcto funcionamiento, desde que librerías de terceros hemos utilizado hasta los sistemas back end, haciendo un estudio de su elección y una explicación de su implementación.

Además se realizará un estudio basado en los estándares de ingeniería del software, realizando un estudio de requisitos, un detalle de las especificaciones, un diagrama de casos de uso etc.

5.1. ANÁLISIS

5.1.1. Identificación del Entorno tecnológico.

En este punto analizaremos las necesidades tecnológicas del proyecto, tanto para el usuario que va a instalar la aplicación como para el programador que lleva a cabo el desarrollo.

5.1.1.1. Dispositivo móvil

Como ya detallamos en el estado del Arte, para la aplicación que se abarca en este proyecto se ha optado por establecer una Api mínima de Android 4.0 debido a:

1. El % de usuarios que tienen una versión anterior del S.O. es pequeño, a la par que lo consideramos un grupo residual, es decir, personas que realmente no hacen un uso intensivo del dispositivo y con unas posibilidades muy reducidas de que quisieran instalarse la aplicación.
2. Las mejoras en cuanto a fiabilidad de la conexión http.
3. Poder omitir muchas trabas por incompatibilidades en versiones antiguas y así reducir uso de la support library.
4. Existen mejoras en el lenguaje que hacen menos tediosa la programación.
5. En la documentación de App42, nuestro proveedor Backend de lógica de juego, se recomienda no usar Apis anteriores a la 4.0.

En base a estas pesquisas que se dan por ejecutar Android 4.0, sobreentendemos que cualquier dispositivo que corra este sistema operativo tendrá potencia suficiente para ejecutar correctamente nuestra aplicación.

5.1.1.2. Entorno desarrollo

Para el entorno de desarrollo he utilizado varios equipos, entre ellos un Mac Mini y un portátil de gama baja, esto me ha sido posible pues la aplicación ha estado subida a un repositorio Git y me ha sido posible mantener el proyecto en los 3 equipos. No obstante en donde se ha llevado a cabo la mayor parte del desarrollo y hubiese bastado en su completitud es en los dos equipos sobremesa que dispongo, los cuales detallo a continuación pues se podría haber realizado de manera única en cualquiera de ellos.

EQUIPO:	Clónico, AMD PHENOM x4-955	Apple Mac MINI
	8Gb ddr2-800	Intel i-5 2500k
	SSD- Samsung Evo 850 500gb.	16gb ddr-3
	AMD Radeon 6970, 1gb.	SSD- Samsung Evo 850 120gb.
		Gráficos integrados.
S.O.	Windows 7	Mac Os
IDE		Eclipse Luna 64bit. Android ADT
Otro Software.		Adobe Photoshop CS6
Simuladores	BlueStack AMD optimized	Android Virtual Device
Dispositivos de desarrollo	Samsung Galaxy s2 i9100 (rooted Android 5.1)	
Dispositivos de Pruebas	ELEPHONE p6000(4.4 kitkat) ELEPHONE p3000(4.4 kitkat) LG I5(4.0.4 ICS) ZTE Nubia z5 (4.4 kitkat)	

Ilustración 61 Tabla recursos Hardware utilizados

Consideraciones Hardware: Hay partes del proyecto que se han desarrollado con un portátil AMD APU e-450, y tras comparar pondría como requisitos mínimos para una experiencia medianamente satisfactoria el contar con un SSD y al menos 8gb de memoria RAM, pues el entorno eclipse Android, consume muchísimos recursos de memoria y hace muchas lecturas del disco duro en un número muy elevado de ficheros distintos, con lo que el SSD al lado de un HD convencional puede establecer una diferencia de entre 15 y 20 veces más veloz en carga y compilado. En cuanto al procesador solo se hace uso de un núcleo por lo que se debería intentar tener un hilo lo más potente posible, no obstante, con el portátil de gama baja AMD, la experiencia de programación, sin llegar a ser completamente satisfactoria, es suficiente para proyectos pequeños.

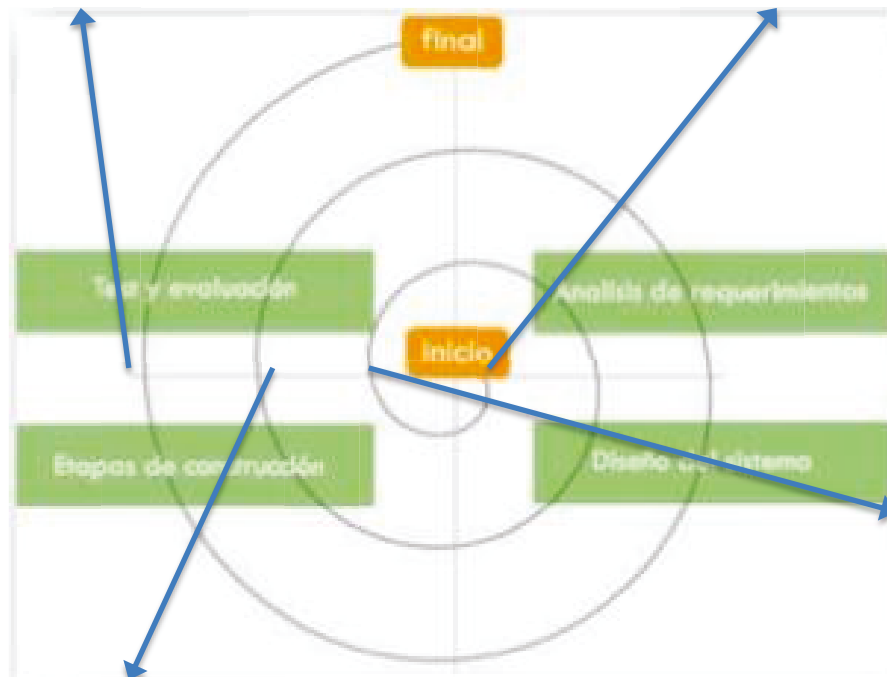
5.1.2. Ciclo de Vida.

Aunque la aplicación se tiene una idea bastante clara desde el principio del prototipo, se ha optado por realizar un desarrollo en espiral basado en un prototipo evolutivo, las principales motivaciones que me han llevado a esta elección han sido:

1. **Complejidad:** debido a la cantidad de módulos que contiene la aplicación y partes integrantes, he optado acometer su desarrollo de forma incremental, lo que nos permite por ejemplo establecer primeramente el sistema para un solo jugador, con todas las complejidades que ello abarca y posteriormente agregarle la capa de multijugador, la de redes sociales y un largo etc.
2. **Prototipado:** Gracias al ciclo de vida, podemos ver carencias en cada fase que pueden generar cambios en otros módulos, con lo que nos permite hacer cambios rápidos en los prototipos que no han sido implementados.
3. **Sistema Funcional por interacción:** Debido a la extensión que puede tener el proyecto, he optado por el desarrollo incremental, permitiéndome tener desarrollos funcionales que me permiten tener un producto acabado que se puede mostrar.

Fase 3: integración con redes sociales y Share de la app

Fase 0: Estudio Viabilidad y concepto.



Fase 1: sistemas básicos, control de usuario, conectividad backend y sistema de juego para 1 jugador.

Fase 2: integración modo multijugador

Ilustración 62 Detalle Ciclo de Vida

5.1.3. Requisitos Software.

A continuación se procederá a detallar los principales requisitos software de la aplicación, esta formulación se llevará a cabo mediante el uso del formato Tabular, proporcionándonos una visión completa del comportamiento esperado de la aplicación. Procedemos a realizar un esquema de la definición de campos.

- **Título:** identificación del requisito
- **Identificador:** Id único de requisito, mediante nomenclatura RSW-XX
- **Tipo:** el tipo de requisito, funcional, rendimiento o usabilidad.
- **Estabilidad:** sensibilidad, si pudiera verse cambiado en desarrollo.
- **Prioridad:** define la importancia del mismo.

REGISTRO DE USUARIOS			
IDENTIFICADOR	RSW-1	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario registrarse mediante email y contraseña o ayudado por Facebook.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

IDENTIFICACIÓN DE USUARIOS			
IDENTIFICADOR	RSW-2	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario identificarse a través del nombre de usuario y la contraseña que previamente haya establecido.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

CONTRASEÑA CIFRADA			
IDENTIFICADOR	RSW-3	TIPO	<input type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input checked="" type="checkbox"/> Usabilidad
DESCRIPCIÓN	La contraseña de usuario debe ser cifrada como mínimo a MDM-5		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

GESTIÓN DE USUARIOS			
IDENTIFICADOR	RSW-4	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario cambiar su perfil, su Fotografía, su email y su contraseña. No obstante el nombre de usuario deberá ser único e inmutable.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

ESTADÍSTICAS DE USUARIO			
IDENTIFICADOR	RSW-5	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario ver sus estadísticas de juego, tanto en partidas de un único jugador como en partidas contra oponentes.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

AMIGOS DE USUARIO			
IDENTIFICADOR	RSW-6	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario ver el listado completo de otros usuarios con los que ambos han aceptado una relación de amistad dentro de la aplicación		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

AÑADIR AMIGOS			
IDENTIFICADOR	RSW-7	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario identificarse añadir a otros usuarios como sus amigos mediante el envío de una petición de amistad.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

RANKING					
IDENTIFICADOR	RSW-8	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario visualizar los ranking tanto totales como entre sus amigos				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

COMPARTIR					
IDENTIFICADOR	RSW-9	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario compartir la app mediante Facebook, WhatAspp, Twitter, Email y SMS.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

COMODÍN EXTERNO A UNA PREGUNTA					
IDENTIFICADOR	RSW-10	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario enviar una pregunta que no sepa a un amigo por : Facebook, WhatAspp, Twitter, Email y SMS.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
	Alta	Media	Baja		

DESCARGA DE PREGUNTAS					
IDENTIFICADOR	RSW-11	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación comprobará en cada inicio para comprobar la versión de base de datos que en caso de no ser la última versión se procederá a su descarga				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

MODO DE JUEGO			
IDENTIFICADOR	RSW-12	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	El usuario podrá elegir entre jugar solo o batir a otros oponentes, tanto amigos como de forma aleatoria.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

LIMITE DE TIEMPO			
IDENTIFICADOR	RSW-13	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	El sistema de juego ocasiona que exista un limite de tiempo para responder a la pregunta, el cual será informado al usuario.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

INDICADOR DE VIDAS			
IDENTIFICADOR	RSW-14	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	El usuario deberá ver claramente en partida e el numero de vidas propias y del oponente, en caso de tratarse de partida multijugador.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

INDICADOR DE NIVEL			
IDENTIFICADOR	RSW-15	TIPO	<input checked="" type="checkbox"/> Funcional <input type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	Las pantallas de partida cambiarán sus colores para ajustarse a cada nivel.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

DIALOGOS INFORMATIVOS					
IDENTIFICADOR	RSW-16	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	El usuario deberá ser notificado mediante un dialogo si gana o pierde la partida, en caso de perderla, se indicará la respuesta correcta, se le debe permitir jugar otra partida o volver al menú.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

DIALOGOS PERSONALIZADOS					
IDENTIFICADOR	RSW-17	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	Los diálogos de información llevarán una casuística de colores distinta según estado. Rojo para preguntas falladas y verde para acertadas.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

CAMBIO DE CONTRASEÑA					
IDENTIFICADOR	RSW-18	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	El usuario podrá deberá poner la contraseña actual antes de permitirse un cambio de esta.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

AUTO INICIO DE SESIÓN					
IDENTIFICADOR	RSW-19	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	Para evitar que el usuario tenga que poner los datos en cada inicio, la aplicación recordará el último usuario validado.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

CIERRE DE SESIÓN					
IDENTIFICADOR	RSW-20	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	El usuario podrá cerrar sesión cuando así lo desee, borrándose todos sus datos de usuario de la app				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

NOTIFICACIONES					
IDENTIFICADOR	RSW-21	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	El usuario deberá ser notificado de los diferentes eventos que ocurran en la app, independientemente de si se esta ejecutando o no.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

INTERFAZ SIMPLE E INTUITIVA					
IDENTIFICADOR	RSW-22	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	El usuario debe ser capaz de manejar la aplicación sin necesidad de realizar un periodo de formación				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

MULTIDISPOSITIVO					
IDENTIFICADOR	RSW-23	TIPO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación permitirá al usuario JUGAR DESDE distintos dispositivos de forma no concurrente sin que ello genere discrepancia de datos.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

TIEMPO DE ESPERA CLIENTE			
IDENTIFICADOR	RSW-24	TIPO	<input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	En caso de un fallo en la comunicación con los servidores no se producirán cambios en la app, el tiempo máximo aproximado deberá estar en torno a los 10 segundos.(3-4 deseables)		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		

DISPONIBILIDAD DE LA APLICACIÓN			
IDENTIFICADOR	RSW-25	TIPO	<input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La disponibilidad de la aplicación estará siempre supeditada a la disponibilidad del backend. En ningún caso corre a nuestro cargo dicha disponibilidad.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		

MAXIMIZAR EL RENDIMIENTO			
IDENTIFICADOR	RSW-26	TIPO	<input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación permitirá utilizará siempre que sea posible procesos en segundo plano para actividades pesadas,		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		

MINIMIZAR LOS BLOQUEOS DE INTERFAZ			
IDENTIFICADOR	RSW-27	TIPO	<input type="checkbox"/> Funcional <input checked="" type="checkbox"/> Rendimiento <input type="checkbox"/> Usabilidad
DESCRIPCIÓN	La aplicación realizará bloqueos de la interfaz de usuario únicamente en aquellas acciones que condicionen la integridad de los datos.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		

CONTROLES ESTANDAR					
IDENTIFICADOR	RSW-28	TIPO	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	En medida de lo posible se intentará utilizar los procedimientos estándar de Android, facilitando al usuario la manejabilidad.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

ADAPTABLE A DIFERENTES PANTALLAS					
IDENTIFICADOR	RSW-29	TIPO	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
			Funcional	Rendimiento	Usabilidad
DESCRIPCIÓN	La aplicación se deberá ajustar a varios tipos de pantalla asegurando una correcta visualización en la gran mayoría de dispositivos.				
ESTABILIDAD	Estable				
PRIORIDAD DEL REQUISITO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
	Alta	Media	Baja		

Ilustración 63 Tablas Requisitos Funcionales

5.1.4. Requisitos de Restricción.

Este tipo de requisitos imponen una serie de restricciones de obligado cumplimiento para llegar a el sistema estable majo el marco que se ha establecido como óptimo para la ejecución de la aplicación.

VERSIÓN MÍNIMA ANDROID			
IDENTIFICADOR	RR-1		
DESCRIPCIÓN	La aplicación se ejecutara únicamente en versiones Android 4.0 y posteriores.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

PERMISOS DE APLICACIÓN			
IDENTIFICADOR	RR-2		
DESCRIPCIÓN	El usuario deberá aceptar los siguientes permisos para poder instalar la aplicación: <ul style="list-style-type: none">○ <i>INTERNET</i>○ <i>ACCESS_NETWORK_STATE</i>○ <i>WRITE_EXTERNAL_STORAGE</i>○ <i>READ_EXTERNAL_STORAGE</i>○ <i>GET_ACCOUNTS</i>○ <i>WAKE_LOCK</i>○ <i>RECEIVE</i>		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

IDIOMA DE LA INTERFAZ			
IDENTIFICADOR	RR-3		
DESCRIPCIÓN	La aplicación se desarrolla en castellano, al igual que el sistema de juego, no se tiene en consideración otras configuraciones de idioma.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

VERSIÓN MÍNIMA ANDROID			
IDENTIFICADOR	RR-4		
DESCRIPCIÓN	La aplicación se ejecutara únicamente en versiones Android 4.0 y posteriores.		
ESTABILIDAD	Estable		
PRIORIDAD DEL REQUISITO	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Ilustración 64 Tablas Requisitos Restricción

5.1.5. Casos de Uso.

El modelo de Casos de Uso nos ofrece un catálogo de las funcionalidades que implementa el sistema, Cada caso de uso representa una sola interacción repetible que un usuario o actor experimenta cuando hace uso de nuestra aplicación.

El diagrama de casos de uso nos sirven para especificar la funcionalidad de la aplicación mediante la interacción con los diferentes actores que existan, ya sean usuarios u otros sistemas. Es una herramienta muy útil, pues detalla de manera sencilla que es capaz de realizar nuestra aplicación.

Se van a desarrollar los casos de uso que se consideran más ilustrativos y establecen un mejor marco entendimiento de las posibilidades globales de la aplicación. Para explicar cada uno de los casos usaremos el modelado de UML 2.0.

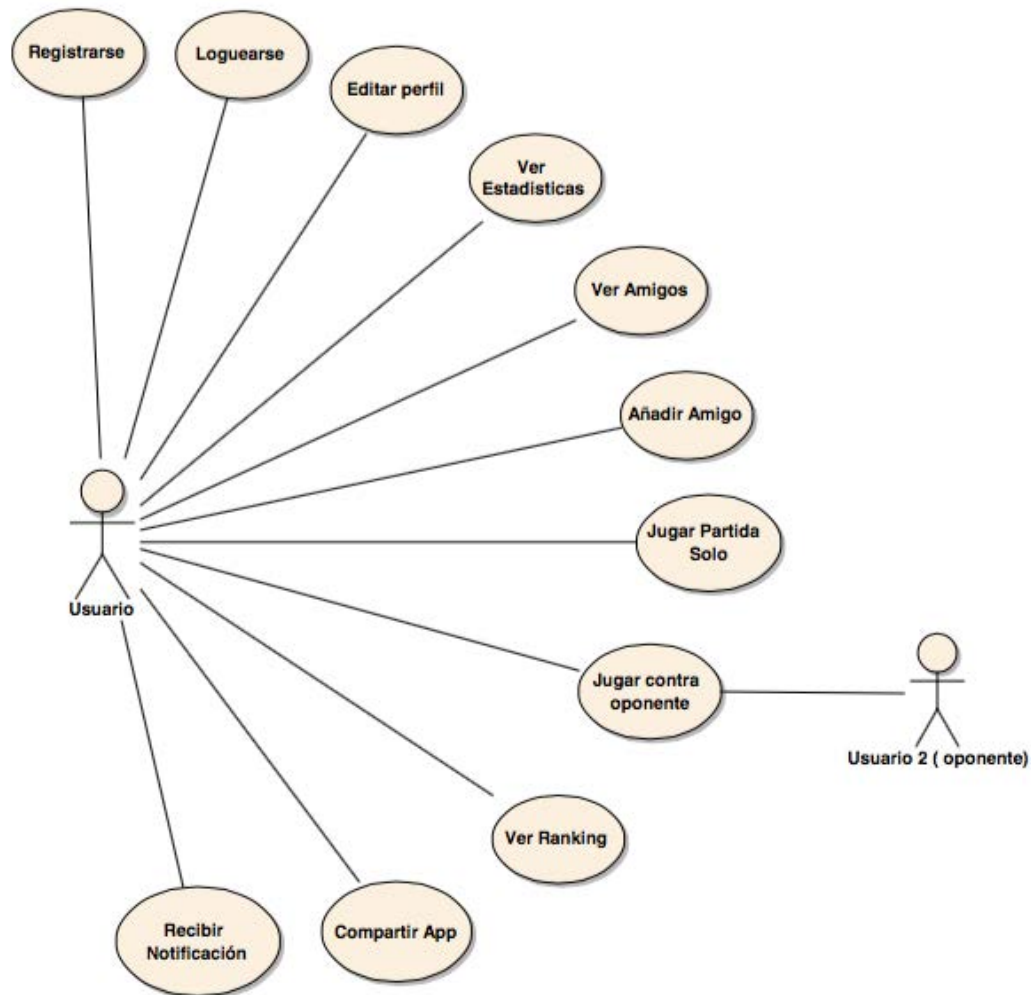


Ilustración 65 Diagrama de Casos de Uso

En todos los casos de uso, excepto registro y logueo se debe incluir el incluye de loguearse, pero para facilitar la esquemática se ha omitido.

REGISTRARSE	
IDENTIFICADOR	CU-1
DESCRIPCIÓN	1. El usuario , en su primer uso se registra en la aplicación.
PRE-CONDICIONES	Conectividad, wifi o móvil

LOGUEARSE	
IDENTIFICADOR	CU-2
DESCRIPCIÓN	1. El usuario accede a la aplicación mediante su usuario y password
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar registrado

EDITAR PERFIL	
IDENTIFICADOR	CU-3
DESCRIPCIÓN	1. El usuario Navega hasta perfil 2. El usuario edita su perfil 3. Elige si hacer cambios y guardarlos o no.
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

VER ESTADÍSTICAS	
IDENTIFICADOR	CU-4
DESCRIPCIÓN	1. El usuario Navega hasta perfil-Estadísticas 2. El usuario ve sus estadísticas
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

VER AMIGOS	
IDENTIFICADOR	CU-5
DESCRIPCIÓN	1. El usuario Navega hasta perfil-Amigos 2. El usuario ve sus Amigos
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

AÑADIR AMIGOS	
IDENTIFICADOR	CU-6
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. El usuario Navega hasta perfil-Amigos 2. El usuario Presiona el botón de nuevo 3. Inserta el nombre del usuario 4. Envía la petición
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

JUGAR PARTIDA SOLO	
IDENTIFICADOR	CU-7
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. El usuario selecciona en el menú jugar Solo 2. Selecciona el nivel 3. Juega la partida 4. Al terminar puede elegir jugar otra partida o salir al menú
PRE-CONDICIONES	<ul style="list-style-type: none"> • Estar logueado

JUGAR CONTRA Oponente	
IDENTIFICADOR	CU-8
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. Navegar hasta jugar contra oponente 2. Seleccionar una partida con turno o en su defecto empezar una nueva 3. Juega la partida 4. Al terminar puede elegir jugar otra partida o salir al menú
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado • Existir un segundo usuario en nuestra misma situación

VER RANKING	
IDENTIFICADOR	CU-9
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. El usuario Navega hasta Ranking 2. El usuario elije entre ver ranking de amigos o general.
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

COMPARTIR APP	
IDENTIFICADOR	CU-10
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. El usuario presiona compartir 2. Selecciona el medio por el que quiere compartir la app 3. El usuario comparte la App
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad, wifi o móvil • Estar logueado

Ilustración 66 Tablas de Casos de Uso

A su vez, existe un segundo actor en nuestro Sistema, el Proveedor de backend de App42, pues este puede iniciar la actividad para enviar una notificación, tendría dos partes, pues por un lado estaría si la notificación es enviada por un segundo usuario por haber generado un evento, y por otro, las iniciadas en el propio backend, para estas tendríamos el siguiente caso de uso:

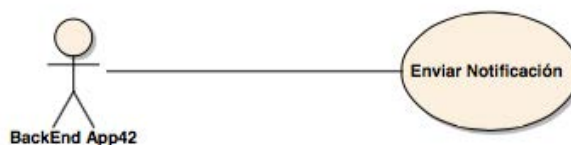


Ilustración 67 Caso de Uso Servidor

ENVIAR NOTIFICACIÓN	
IDENTIFICADOR	CU-11
DESCRIPCIÓN	<ol style="list-style-type: none"> 1. El servidor envía una notificación 2. El usuario la recibe y puede elegir abrir la app
PRE-CONDICIONES	<ul style="list-style-type: none"> • Conectividad cliente final • Poseer una cuenta asociada.

Ilustración 68 Tabla Casos Uso Servidor

5.1.6. Diagramas de secuencia

Con los diagramas de secuencia se muestra la interacción del conjunto de objetos pertenecientes a la aplicación a lo largo del tiempo para cada uno de los casos de uso.

CU1- REGISTRO: he optado por incluir en este diagrama de secuencia la operativa de descarga de las nuevas palabras pues al registrar el usuario esto se realizará siempre la primera vez. En caso de que la base de datos ya estuviese, ese paso sencillamente se omitiría.

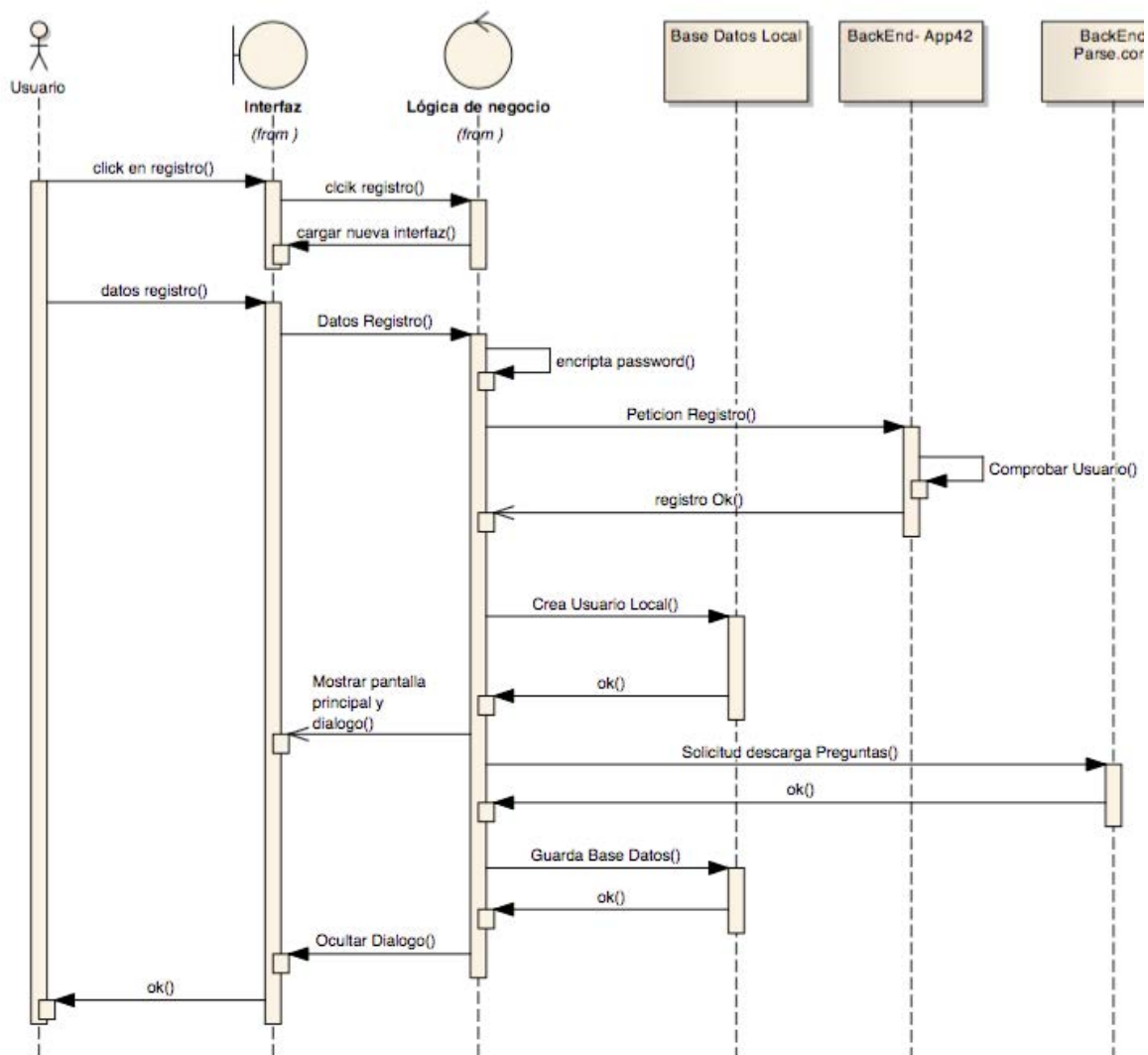


Ilustración 69 Diagrama de secuencia CU1

CU2- LOGUIN USUARIO:

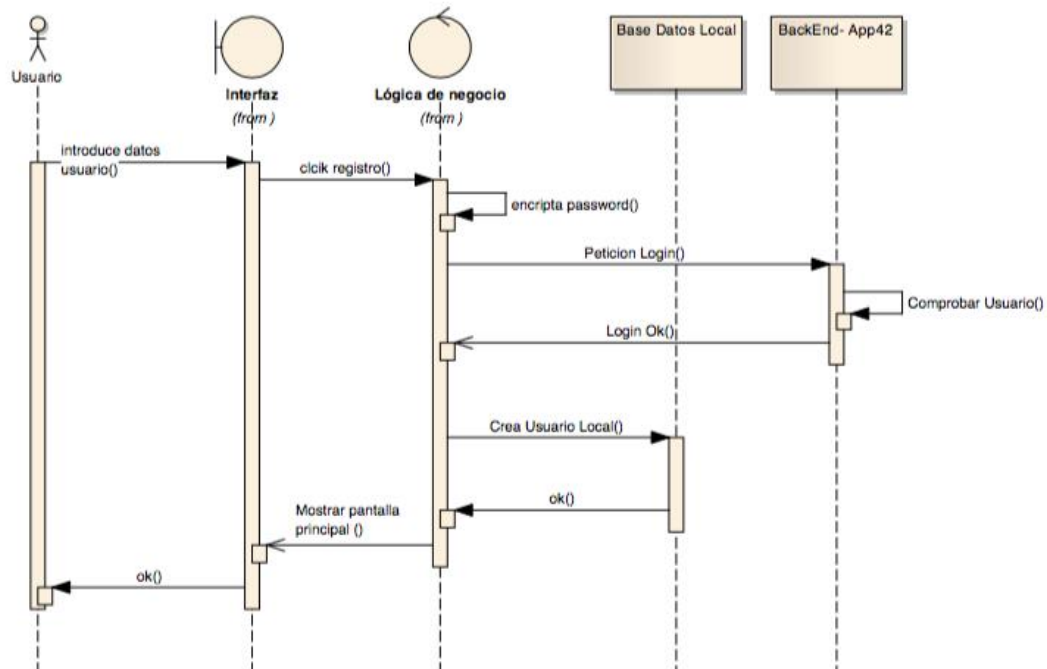


Ilustración 70 Diagrama de secuencia CU 2

CU3- EDITAR PERFIL:

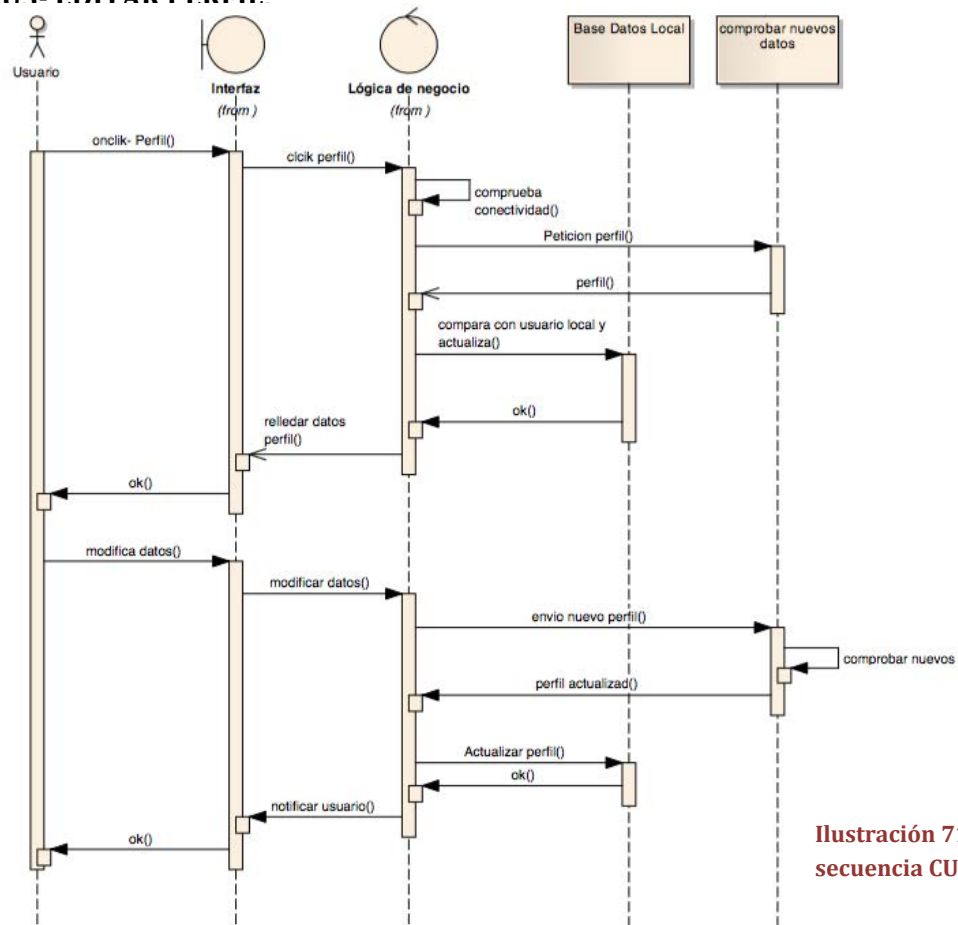


Ilustración 71 Diagrama de secuencia CU 3

CU4- VER ESTADISTICAS:

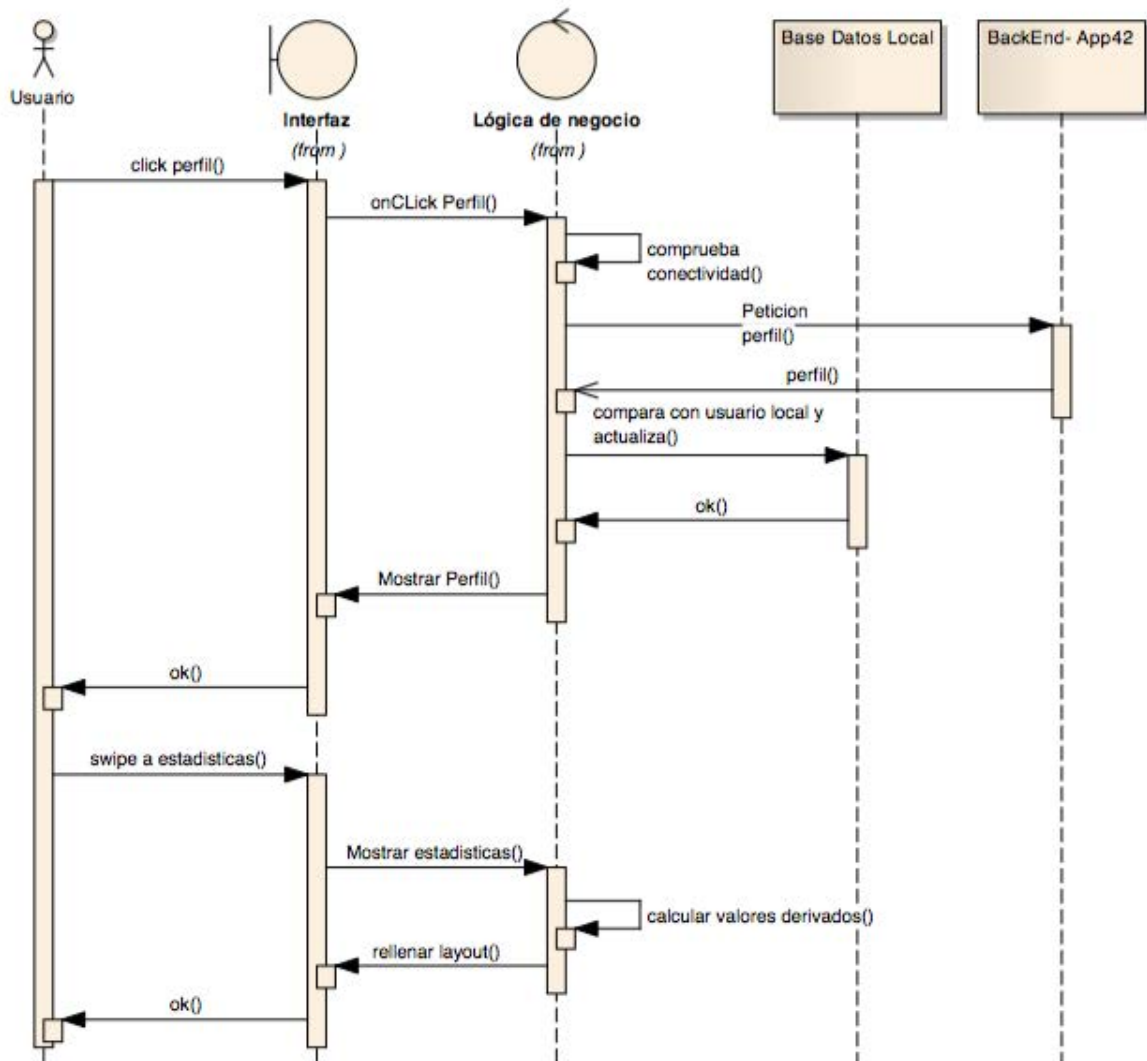


Ilustración 72 Diagrama de secuencia CU 4

Para llegar a las estadísticas hace falta pasar por toda la lógica referente al perfil, pues se trata de un ViewPager con varias páginas, por lo que se cargará también el perfil.

CU5- VER AMIGOS:

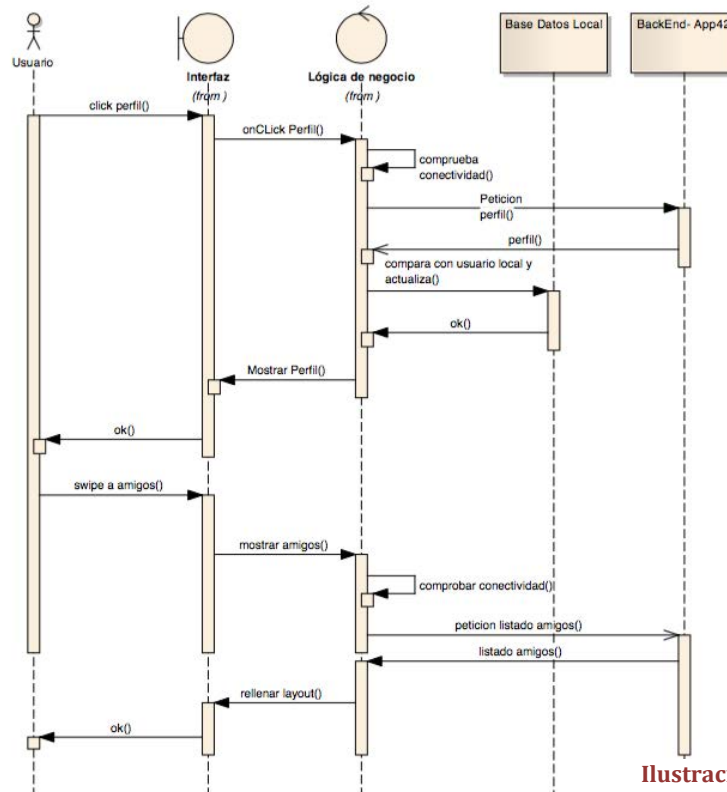


Ilustración 73 Diagrama de secuencia CU 5

CU6- AÑADIR AMIGOS:

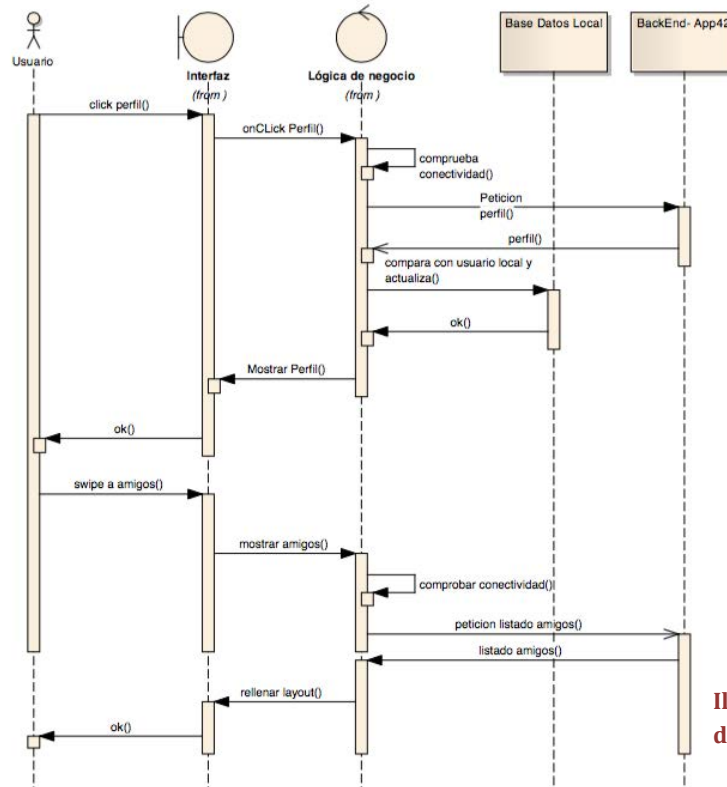


Ilustración 74 Diagrama de secuencia CU6

CU7- JUGAR SOLO:

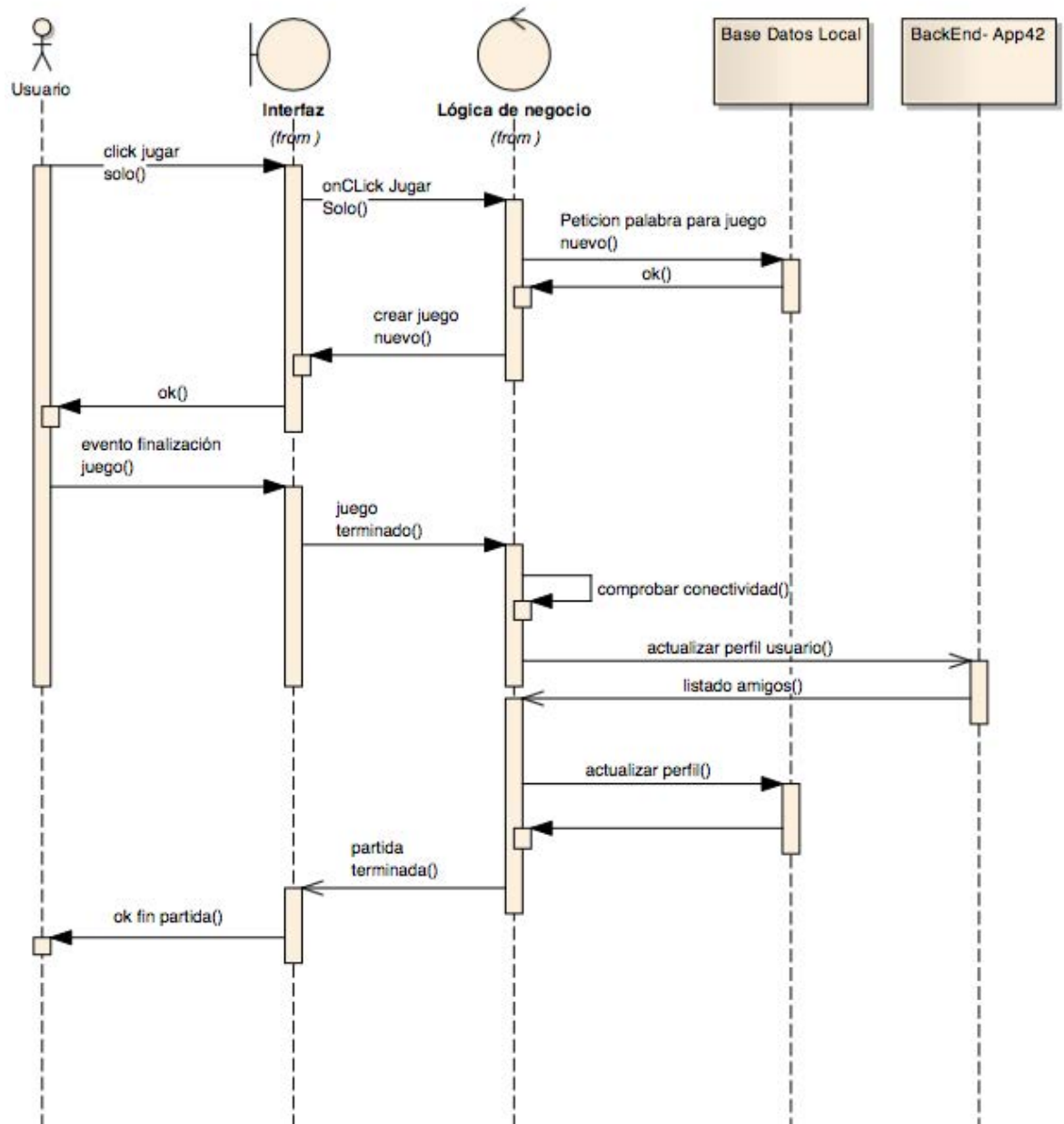


Ilustración 75 Diagrama de secuencia CU 7

CU8- JUGAR CONTRA Oponente:

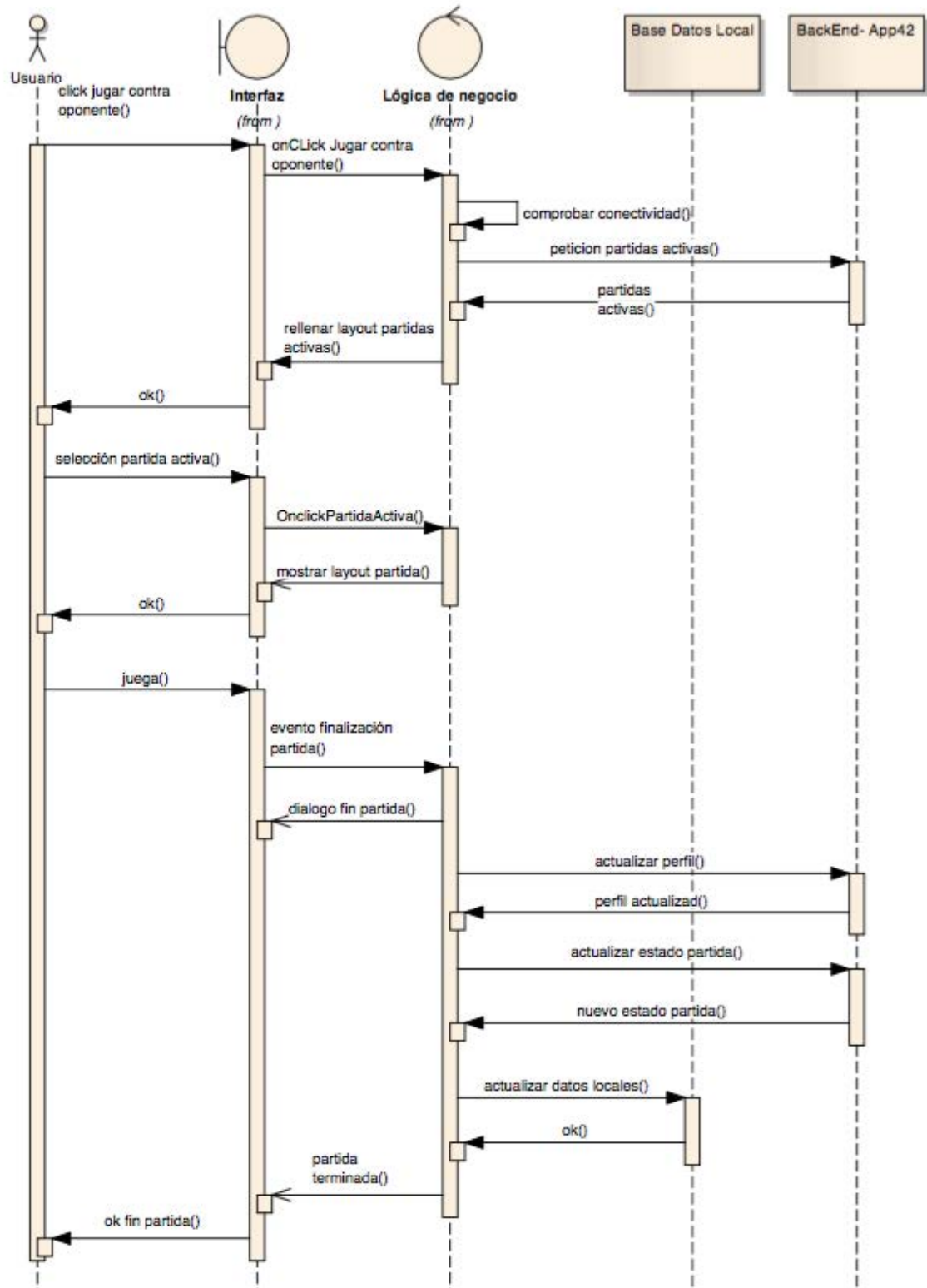


Ilustración 76 Diagrama de secuencia CU 8

CU9- VER RANKING:

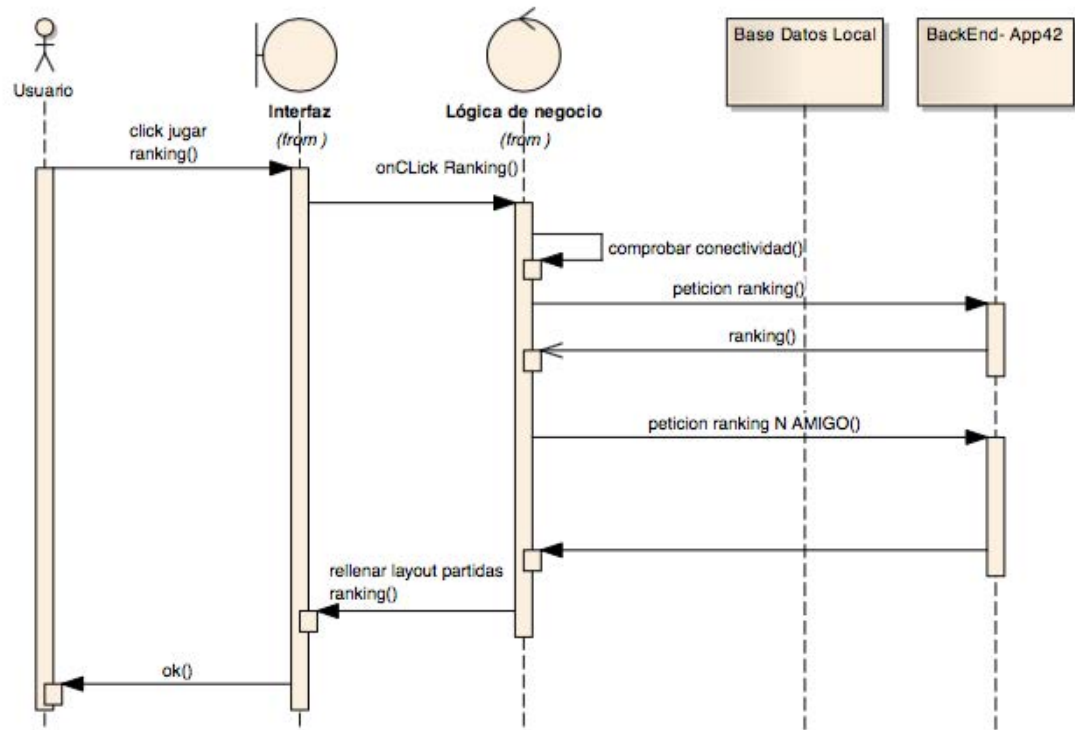


Ilustración 77 Diagrama de secuencia CU 9

CU10- COMPARTIR:

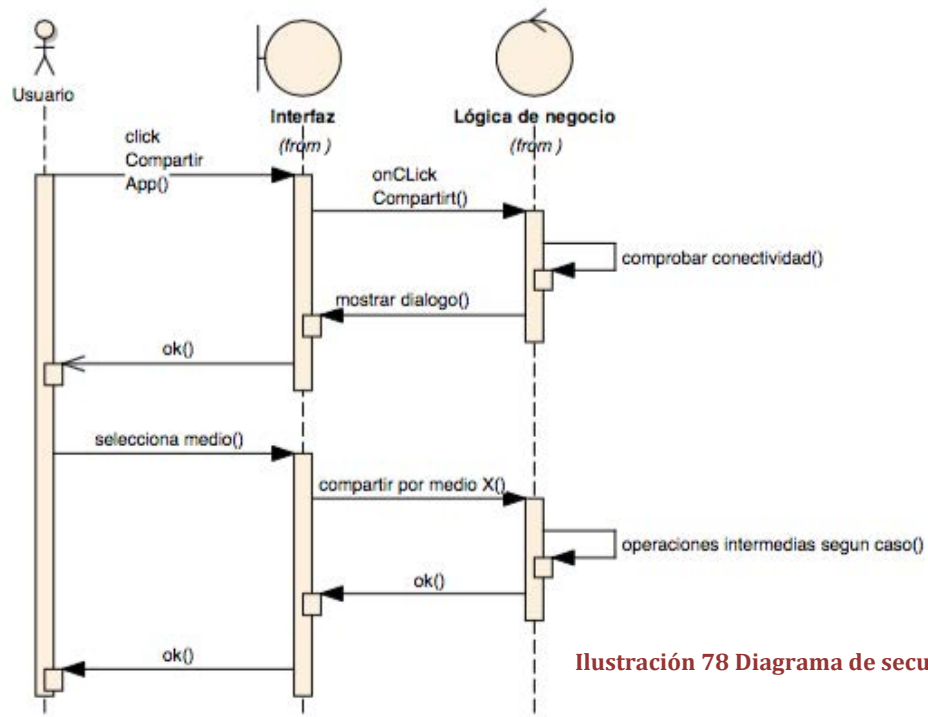


Ilustración 78 Diagrama de secuencia CU 10

5.2. DISEÑO

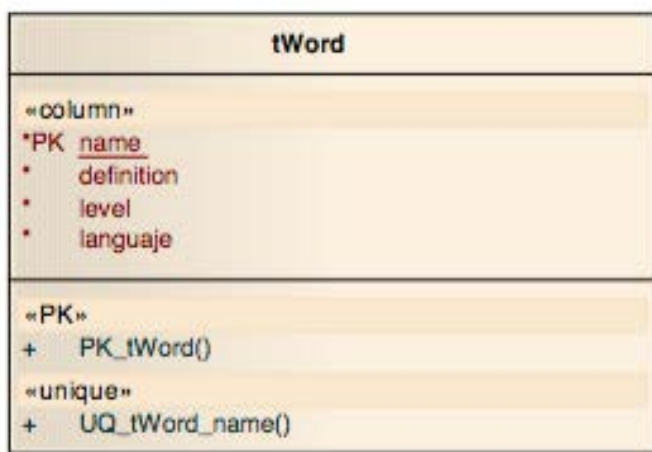
Partiendo del estado del arte descrito, es decir saber con que recursos contamos y teniendo en cuenta todos los requisitos y especificaciones que se han ido detallando en apartados anteriores se procede a formalizar el diseño del sistema, desde el origen de los datos hasta la estructura de la interfaz.

5.2.1. Modelo de datos.

Para el correcto uso de la aplicación, se tienen que utilizar diferentes modelados de datos según el ámbito sobre el que estemos tratando. Encontramos 2 bloques, los correspondientes a la lógica de las preguntas y los correspondientes a la lógica de juego que nos obliga el Backend.

5.2.1.1. Datos en Local - SQLite

La Base de datos con las preguntas del juego es muy sencilla, se trata de una simple tabla con 4 campos, esta está alojada en SQLite de forma local en el dispositivo, ahorrándonos tener que acceder a la red cada vez que ejecutemos una partida.



Nombre: Contiene la respuesta a la pregunta, es primaryKey y así nos aseguramos que no se repite la palabra.

Definition: pregunta a realizar.

Level: un entero con el nivel, de 1 a 5.

Lenguaje: actualmente todo es “es”, se modeló así para dejar la puerta abierta a implementar una diferenciación de idiomas.

Ilustración 79 Modelo de Datos SQLite Local

Esta Base, existe con estos mismos campos en el Servidor de Parse.com pues es el lugar de donde se nutre la aplicación para generas esta base de datos. En Parse existe otra tabla muy sencilla que únicamente cuenta con el campo de id de base de datos, el cual actualizaremos cada vez que cambiemos las palabras y la aplicación por si sola se encargará de regenerar la base de datos local.

Como puede darse el caso de que falle la conectividad, además de ser datos de acceso recurrente, se ha creado una tabla usuario en el SQLite local. Esta tabla se nutre de APP42 y posee el siguiente esquema:

tUsuario
«column»
name
surname
idFacebook
email
password
country
sex
score
totalIndividualGames
TotalIndividualWins
totalMultiplayerGames
TotalMultiplayerWins
profileImageUrl
profileImageUrl
ProfileImageBit

Ilustración 80 Modelo Datos Usuario

Esta tabla la utilizaremos para mostrar rápidamente los datos en pantalla mientras esperamos la petición del servidor, o si se estableciera un error de conectividad poder ver el usuario y acceder a sus datos. No obstante, como la aplicación puede sincronizarse debido a el uso de un mismo usuario en diversos dispositivos, estos datos se comprobarán en cada acceso y serán modificados si han sufrido una actualización.

5.2.1.2. Modelo Datos en Backend App42

Debido al funcionamiento de App42, que guarda Json, sobre bases de datos, he realizado un modelo de datos lógico sin preocuparme de cómo son estos guardados internamente en el backend.

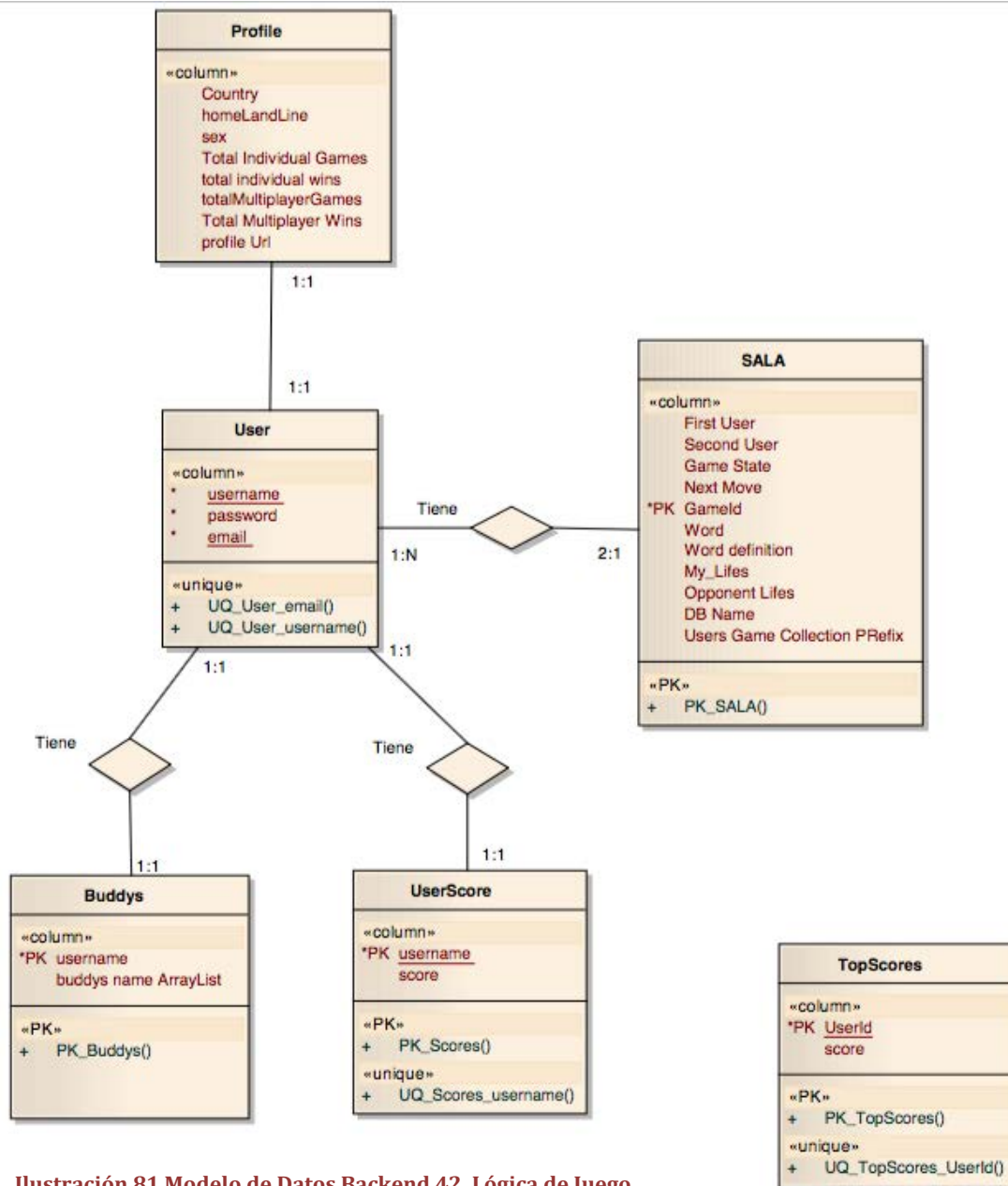


Ilustración 81 Modelo de Datos Backend 42, Lógica de Juego

Para acceder a este modelo de datos en App42, cada tabla pertenece a lo que ellos llaman un servicio, el userService gracias a el que podemos recuperar las tablas de user y profile, el buddyService que pasándole el parámetro del Id nos devolverá el listado de

nombres de amigos, el ScoreService que nos dirá la puntuación multijugador del usuario y que guardaremos para que estos datos sean accesibles desde cualquier cuenta de usuario, pudiendo llenar los Rankings.

De Manera autónoma y no vinculada a otras tablas tenemos la tabla de máximas puntuaciones, la cual es accedida de igual manera por ScoreService y nos devuelve un Json con las tuplas de la tabla, Id y puntuación. Esta tabla es autogenerada por el servidor.

Nótese que para el sistema de puntos de un solo jugador no se podía implementar utilizando el score service pues ya estaba ocupado por multijugador, además esta información no debe ser accesible más que por el propio usuario por lo que se ha optado por crearla en el backend en lo que app42 establece como profile.

5.2.2. Esquema Aplicación; Navegación de pantallas y activitys.

A continuación, como parte fundamental del diseño se establecerá la jerarquía de navegación por pantallas, organizándola por niveles y funcionalidades. Para llegar a este diseño final se han dechado numerosos prototipos en papel y juegos de prueba realizados en entorno controlado para obtener la mejor interacción posible entre usuario-aplicación.

Se han utilizado fragmentos como Base de toda la aplicación, por lo que nuestro número de actividades se ve reducido a dos.

- **SplashActivity:** esta actividad comprueba si ya existe usuario sin tener que cargar complejos layouts , según su existencia procederá a invocar una actividad u otra.
- **LoginActivity:** Si el usuario no estuviese registrado o logueado, esta actividad es llamada, posee todos los componentes necesarios para realizar el alta / logueo de usuario.
- **MainActivity:** Esta es la actividad principal en donde se desarrolla toda la aplicación, contiene toda la lógica de juego, su diseño es modular basado en fragmentos y hace uso de gestos típicos de Android como puede ser swipes, un ViewPager etc.

ESQUEMA DE NAVEGACIÓN DE PANTALLAS



Ilustración 82 Esquema de Pantallas y actividades

5.3. IMPLEMENTACIÓN

En esta sección se procede a explicar de forma detallada los aspectos relevantes e importantes que se han tenido que utilizar para llevar a cabo la implementación de la aplicación. Se omiten funcionalidades básicas de Android que se han detallado en capítulos anteriores como pueden ser las carpetas, los layouts utilizados. Se detallarán cada una de las tecnologías y añadidos sobre los que funciona la app. También se ha realizado en este apartado, en los sub-apartados de corresponda un proceso de selección que estudia el porqué hemos utilizado la implementación de cada herramienta, en caso de existir varias coexistiendo.

5.3.1. Sentando las bases del desarrollo incremental- GIT



Para llevar un control del desarrollo, poder realizarlo en múltiples equipos sin que ello suponga una ardua y tediosa tarea y sobre todo, poder llevar un control de versiones que en caso de un mal desarrollo tener un punto de restauración cercano, el desarrollo del proyecto se ha realizado utilizando un repositorio de tipo GIT , hospedado en Bitbucket, la decisión de utilizar Bitbucket es a parte de por tener ya una cuenta antes de empezar con el proyecto, porque nos ofrece una capa gratuita que cumple ampliamente las necesidades del proyecto como son la posibilidad de participar hasta 5 desarrolladores, en el caso que nos acata va a tratarse de sólo 1 y por otro lado, el repositorio que creamos podemos hacerlo privado, no estando obligados a compartirlo con la comunidad en caso de que no lo deseemos.

Para realizar las operaciones sobre el repositorio hemos utilizado la herramienta SourceTree, que nos permite, mediante su interfaz gráfica , interactuar con el repositorio de una manera clara y sencilla.

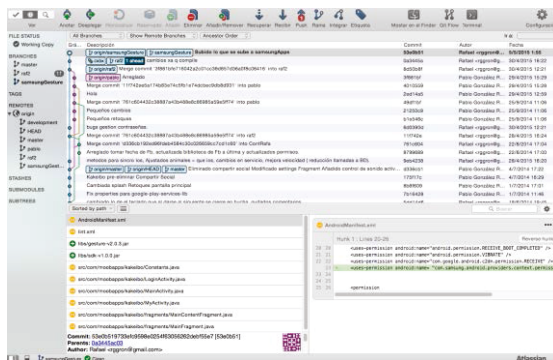


Ilustración 83 Detalla Vista Source Tree

El proyecto consta aproximadamente de unos 165 Commits y se han creado Ramas para cada nueva funcionalidad, Manteniendo una rama Maestra con la versión con los elementos finalizados y una nueva subrama con cada implementación incremental.

5.3.2. Sistemas Backend.

En este apartado vamos a repasar las diferentes soluciones Back-end que encontramos en el mercado, estudiaremos que necesidades tiene nuestra aplicación y obtendremos como conclusión el porque el uso de las finalistas es la mejor solución. Empezare con los Backend que hemos descartado en fases más tempranas y porqué.

Se dejan Fuera los Backend de Amazon y Azure para cualquier cometido.



Amazon, actualmente provee uno de los más grandes y completos Backend que podemos encontrar en el mercado. Ofrece infinitas posibilidades, desde la creación de servidores virtuales, autoescalables con multitud de opciones de cacheado, balanceadores, creación de replicas en diferentes lugares del mundo para optimizar los tiempos de acceso según donde se encuentre el usuario y un casi ilimitado etc.

Entre todos los servicios que ofrece Amazon, los que necesitaríamos utilizar para nuestra aplicación son básicamente Amazon Dynamo DB, que nos permite crear tablas en donde gestionaríamos las partidas, o por otro lado Amazon S3, si fuésemos a utilizar ficheros XML a modo similar que hace APP42.

Entre los motivos por lo que Amazon ha sido mi primera opción en ser desechada es que es una herramienta que no esta orientada a programadores Mobile, Amazon esta orientado a TIC, empresas que vayan a ofrecer grandes capacidades de gestión en la nube y aunque hacerse con la consola es un proceso que es asumible para un informático, el proceso de aprendizaje y la infinidad de opciones que no nos son de utilidad para nuestro pequeño juego online, suponen un tiempo de aprendizaje y despliegue que no iría en consonancia con los requisitos de la APP. Por otro lado Amazon sólo ofrece una capa gratuita por un tiempo de 1 año, pasado este, aunque el precio es muy bajo nos obliga a tener asociada una tarjeta para realizar los pequeños cobros mensuales.



Windows Azure, al igual que Amazon, es un backend genérico que ofrece multitud de servicios que nos quedan excesivos para el cometido de la aplicación. No obstante, existe una capa personalizada de Azure denominada "Servicio de Aplicaciones" que nos ofrece de manera gratuita un backend para hasta 10 aplicaciones diferentes. Entre los principales problemas que encontramos a Azure se encuentran:

- No es un backend optimizado para juegos. Con lo que podríamos guardar las tablas, pero tendríamos que hacer una lógica de consultas que ralentizarían la programación.
- Esta limitado a 500 dispositivos al día y a 500 llamadas estándar, lo que se nos antoja que puede quedarse limitado en un tiempo muy reducido.
- La documentación no es clara para desarrolladores Mobile.

5.3.2.1. Backend de juego - APP 42

Se ha optado por utilizar el concepto de Back-end as a service, pues existen muchas soluciones en el mercado que para la finalidad de la aplicación cumplen con creces las necesidades, además nos libran de tener que realizar un servidor permitiéndonos centrarnos en otros aspectos más importantes.



Parse nos ofrece un Backend especialmente optimizado para las plataformas Android e IOS, además de poder integrarse con web, servidores y un largo etc. Posee una capa gratuita que nos sería suficiente para todo el contexto de aplicación además de contar con una documentación muy elaborada y sencilla de utilizar. No obstante Parse no es un backend especializado en juegos y no nos ofrece la sencillez para una rápida gestión de partidas y jugadores, con lo que al igual que con Windows Azure, nos obligaría a programar más lógica de juego basada en las consultas a tablas.



GAME SPARKS

GameSparks sí es un backend especializado en Apps y juegos, entre sus principales características:


 GAME SPARKS	Transferencia de datos	Notificaciones Push	Api-Calls	Espacio Almacenamiento
Características Capa gratuita	20GB/mes	20m / total	20m/total	25GB
Ventajas		➤ Uso de todas sus características desde el momento inicial.		
Inconvenientes		➤ No ofrece soporte en la capa gratuita. ➤ Una vez que la aplicación se lanza, se cobra a 2 cent/ usuario. ➤ No tiene Api Nativa		
Calidad documentación		Aunque posee muchos videos y tutoriales explicativos, no ofrece listados de fácil acceso a todas las características, es engorroso encontrar lo que buscamos.		

Ilustración 84 Tabla Características Game Sparks

El inconveniente de cobrar 2 céntimos por cada nuevo usuario que se registre es motivo único y suficiente para descartar el Backend.

brainCloud	Transferencia de datos	Notificaciones Push	Api-Calls	Espacio Almacenamiento
Características Capa gratuita	-	20m / mes	100 por usuario y día	25GB
Ventajas		<ul style="list-style-type: none"> ➤ Api Nativa (según ellos) ➤ Posee un dashboard claro y sencillo, con mucha información. 		
Inconvenientes		<ul style="list-style-type: none"> ➤ Máximo 100 usuarios y 100 apicalls por usuario al día. ➤ Me ha sido totalmente imposible encontrar documentación específica para Android, solo hay cocos 2d, y JavaScript, aún de que en su página principal dicen poseer SDK Android. 		
Calidad documentación		NULA		

Ilustración 85 Tabla Características Brain Cloud

La imposibilidad de encontrar una documentación clara, a parte de la restricción del numero de apicalls y usuarios tan limitada nos obligan a desechar el Backend.



Finalmente nos encontramos con App42-Cloud Api, Un Backend especialmente diseñado para juegos en donde la gestión de usuarios y partidas esta mu conseguida. Funciona mediante la creación de llamadas “salas” que no son más que archivos JSON, de usuarios, partidas etc. que se van creando según lo que el usuario realice. Por ejemplo, si un usuario crea una partida nueva retando a un oponente , se crearan dos JSON con los datos de la palabra solución el turno, las vidas de cada adversario, este JSON es muy fácilmente recuperable mediante la API nativa tanto para Android como para iOS.

Una de las cosas que más me ha gustado de este Backend es la facilidad de creación de usuarios y documentos, pues es totalmente transparente. No hay que definir tablas ni campo, cada vez que hacemos una petición de envío de una partida o de un usuario, el backend automáticamente genera los documentos necesarios.

La única preconfiguración que hay que realizar es definir el nombre de la app, configurar las notificaciones como se detallará más adelante y ajustar los permisos y estado de la app.


	Transferencia de datos	Notificaciones Push	Api-Calls	Espacio Almacenamiento
Características Capa gratuita	ilimitada	1M/mes	1M/mes	1GB
Ventajas	<ul style="list-style-type: none"> ➤ Api Nativa ➤ Posee un dashboard claro y sencillo, con mucha información. ➤ Soporte vía email de rápida respuesta ➤ Funciona mediante “salas” no hay casi configuración previa a la hora de crear la app. 			
Inconvenientes	<ul style="list-style-type: none"> ➤ La limitación de espacio a 1Gb ➤ La Api Asíncrona es muy nueva y todavía posee ciertos fallos , he tenido que contactar con ellos para solucionarlos. ➤ Tiempos de respuesta algo elevados y baja tasa en transferencia de datos. 			
Calidad documentación	EXCELENTE.			

Ilustración 86 Tabla Características App 42

Finalmente APP42-CLOUD API, ha sido la seleccionada por su claridad en documentación, además de ofrecernos una capa gratuita sin limite de tiempo que se nos antoja suficiente en casi todos sus ámbitos. App 42 nos ofrece un SDK de desarrollo en el que podemos utilizar los siguientes servicios:

App42 BaaS/MBaaS Android SDK supports following services

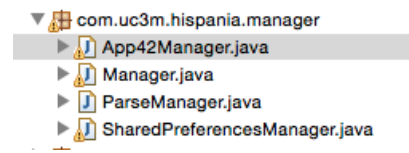
- User Service
- Push Notification Service
- Timer Service
- Buddy Service
- Leaderboard Service
- Gallery Service
- Review Service
- Message Service
- Logging Service
- Storage Service
- Event Service ^{NEW}
- Social Service
- Avatar Service
- Reward Service
- Geo Service
- Cart Service
- Recommender Service
- Custom Code Service
- Gift Management Service
- A/B Test Service
- Achievement Service
- Upload Service
- Session Service
- Catalogue Service
- Email Service

Para el desarrollo de esta aplicación se ha utilizado:

- User Service: Nos permite gestionar las altas y bajas de usuario, así como su registro y modificación de datos.
- Push Notification Service: Se utiliza para avisar a otros jugadores sobre eventos relevantes, nuevas peticiones de amistad, cambios en partidas...

- Buddy Service: nos permite, de una manera sencilla crear relaciones entre diferentes jugadores, pudiendo crear así relaciones de amistad para ver las puntuaciones de los amigos, poder jugar partidas accediendo a ellos directamente...
- Session Service: Crea una sesión para el usuario identificado, permitiendo el acceso a los datos del backend.
- StorageService: Este es el servicio sobre el que más uso se hace, crea un documento Json vinculado a una serie de prefijos que nos permiten guardar en una base de datos los datos de las partidas, el usuario...
- ScoreService: Permite gestionar las puntuaciones, haciéndolas visibles y permitiendo guardar y recuperarlas con todos los usuarios.

Para la **utilización de los servicios** de APP42 se ha desarrollado un manager, denominado APP42Manager en donde se engloban todas las funciones y que son necesarias para el correcto uso de la aplicación, a la vez que todo lo necesario para un correcto manejo de la conectividad en segundo plano. Esta clase podríamos separarla en X bloques:



```
// Singleton
public static App42Manager defaultManager(Context context) {
    if (mInstance == null) {
        mInstance = new App42Manager(context);
    }
    return mInstance;
}

private App42Manager(Context context) {
    App42API.initialize(context, Constants.APP42_APP_KEY,
        Constants.APP42_SECRET_KEY);
    this.userService = App42API.buildUserService();
    this.socialService = App42API.buildSocialService();
    this.sessionService = App42API.buildSessionManager();
    this.storageService = App42API.buildStorageService();
    this.buddyService = App42API.buildBuddyService();
    this.gameService = App42API.buildGameService();
    this.scoreBoardService = App42API.buildScoreBoardService();
    this.pushNotificationService = App42API.buildPushNotificationService();
    this.imageProcessorService = App42API.buildImageProcessorService();
    this.uploadService = App42API.buildUploadService();

    // App42CacheManager.setPolicy(Policy.NETWORK_FIRST);
    // App42CacheManager.setExpiryInMinutes(1);
    App42API.setOfflineStorage(true);
}
```

➤ Un primer bloque con la instanciación de un singleton para la clase App42Manager , y la inicialización en el constructor de las Api y de los diferentes servicios que vamos a utilizar.

- Un 2º Bloque con los métodos referentes al Usuario:
- getCurrentUser
 - CreateUser
 - CreateOrUpdateUserProfile
 - UpdateUserEmail
 - UpdatePassword
 - UpdateUserfromCurrentUser
 - UpdateUserfromAUser
 - authenticateUserWithUserName
 - getUserWithUserName

- un 3º bloque con todos los métodos de sesión
 - authenticateSessionForCurrentUser
 - authenticateSession
 - getCurrentSession
 - logoutCurrentSession

- 4º Bloque con métodos de guardado de ficheros JSON (todo lo referente a la lógica de juego multijugador)
 - createGameVersus
 - getUserGameList
 - getUserActiveGameList
 - getUserFinishedGameList
 - updateGameFromJson
 - removeGame
 - createGameForRandomPlayer
 - removeRandomGameRequest
 - checkGameForRandomPlayer

- 5º: métodos de “amigos”
 - getAllMyFriendRequest
 - getAllFriendRequestFrom
 - getAllMyFriends
 - getAllFriendsFrom
 - acceptFriendRequestFrom
 - rejectFriendRequestFrom
 - sendFriendRequestTo

- un 6º Bloque con los métodos de envío de notificaciones, como se verá mas adelante, la recepción se hace mediante otra clase específica.
 - registerUser: este método registra el token de usuario para las notificaciones.
 - unregisterUser
 - notifyUsername: manda una notificación a el usuario que le especifiquemos.

- 7º Bloque de métodos de Puntuación:
 - saveUserScore
 - getTopScores
 - getTopFriendsScores
 - getScoresByUser
 - getHighestScoresByUser

Todo método que utilice red está ejecutado en un hilo separado, a su terminación, es devuelto al hilo principal y llamado al callback para continuar con la ejecución.

5.3.2.2. Backend de datos e imágenes - Parse.com

Para manejar la base de datos con las preguntas se ha optado por utilizar un backend separado a el de la lógica de juego. Esta decisión ha sido tomada en fase de diseño tras el estudio de las fortalezas y debilidades de APP42, la separación repercute en una serie de ventajas claras:

- Descargar los datos e imágenes de usuario consume un alto número de llamadas y de ancho de banda en comparación con la lógica de juego. La separación de estas, restará uso al Backend de juego sobre todo si decidimos actualizar la Base de datos, nos creará flujos que harán que sea más fácil en momentos puntuales llegar a los límites de la capa gratuita.
- APP42 no posee un modelo claro de tabla para guardas los datos, podríamos crear un Json y subirlo, pero nos obligaría a subirlo entero cada vez que quisiésemos actualizar la base de datos.
- APP42 no ofrece herramientas para grandes volúmenes de datos desde la consola, con lo que nos obligaría a crear una aplicación de gestión de estas subidas.
- En una primera Fase se implementó usando App42 como backend de imágenes pero los tiempos de respuesta eran muy lentos y se hacía muy notable la aparición de las imágenes a los 4 o 5 segundos de abrir la pantalla, Este efecto se veía compensado con el uso de UniversalImageLoader, pero ya que se iba a utilizar otro backend para los datos, las modificaciones no eran excesivas para cortar de raíz este problema.

La elección del backend aquí era clara, únicamente necesito una forma fácil y potente de poner una tabla con datos para su replicación en el SQLite local. Con esta premisa nos centramos en los Backend más grandes y potentes, buscamos la potencia y el tiempo de respuesta pues el manejo de la imagen requiere más flujo de datos, con lo que los backend más pobres resultan en un mayor tiempo de respuesta. Finalmente nos quedamos con AMAZON, AZURE Y PARSE.COM.

Los dos primeros los descartamos por el mismo motivo que se detallo anteriormente, saliendo vencedor Parse.com por los siguientes motivos.

Parse	Transferencia de datos	Notificaciones Push	Api-Calls	Espacio Almacenamiento de ficheros	Espacio Almacenamiento de Base de datos
Características Capa gratuita	2Tb/Mes	1M/mes	30 por segundo prorrateadas.	20GB	20GB
Ventajas		<ul style="list-style-type: none"> ➤ Api Nativa ➤ Posee un dashboard claro y sencillo, con mucha información y en donde es posible la interacción directa. ➤ Esta montado sobre una infraestructura Amazon, con lo que es rápido y potente. ➤ Ofrece poder hacer 1 Job en Base de datos, muy útil para por ejemplo hacer un recolector de basura en fotos de perfiles que ya no se usen. ➤ Potente herramienta de Analytics. ➤ 20 Gb de ficheros que en este caso serán destinados a Fotografías de los perfiles, nos eliminan completamente el cuello de botella de App42 con su único Gb ➤ 20Gb para bases de datos. ➤ Es posible subir ficheros CSV y JSON para ser recreados como base de datos 			
Inconvenientes					
Calidad documentación		EXCELENTE.			

Ilustración 87 Tabla Características Parse

Para integrar Parse ⁶³ en nuestra aplicación y por tener un SDK de desarrollo propio, lo primero que se ha hecho es integrar las librerías de dicho SDK. La configuración de Parse en el dashboard es sencilla, se crea desde el dashboard un nombre de la aplicación y obtenemos un apikey y un secret Key.

Para la utilización de Parse, al igual que en situaciones anteriores he optado por crear un manager que gestione toda la operatividad con este Backend. La clase recibe el nombre de "ParseManager", crea el singleton que será inicializado en "MyAplicación" y desde el manager tendremos los métodos con los que trabajaremos y el manejo de dicho singleton.

Para trabajar con Parse hay que entender una serie de conceptos básicos sobre los que se basa su estructura.

- **El Objeto PARSE :** Trabajar con los datos en Parse se realiza mediante el “parseObject” , es un Json de claves valor sobre el que ya existen métodos predefinidos para obtener los datos. Sobre estos Objetos podemos:
 - **Añadir campos:** objeto.put
 - **Recuperar campos:** objeto.getString , objeto.getBoolean etc....
- **Querys Parse:** Se utilizan para recuperar Arrays que contienen los objetos que cumplen la query.
 - `ParseQuery<ParseObject> query = ParseQuery.getQuery("Questions");`
Esta recuperará todos los objetos de la tabla Questions.
- **Subir Objetos a Parse:**
 - En el caso de querer subir sólo un registro a la tabla, crearemos el objeto y lo subiremos a la tabla correspondiente.

```
ParseObject bigObject = new ParseObject("BigObject");
bigObject.put("myNumber", myNumber);
bigObject.put("myString", myString);
bigObject.put("myDate", myDate);
bigObject.put("myData", myData);
bigObject.put("myArray", myArray);
bigObject.put("myObject", myObject);
bigObject.put("myNull", JSONObject.NULL);
bigObject.saveInBackground();
```

* Nótese como todos los métodos existen en síncrono y asíncrono.

- en el caso de tratarse de un fichero, como es el caso de nuestro aplicativo, primero se deberá subir el fichero con la imagen, y posteriormente asociarlo en una tabla.

```
private void uploadImageToParse(String afileName, String description,
    byte[] fileData, String userName,
    final ManageUploadedCallback uploadedCallback) {
    final String newTitle = userName + afileName;

    // Create the ParseFile
    final ParseFile file = new ParseFile(newTitle, fileData);
    // Upload the image into Parse Cloud
    file.saveInBackground();

    // Create a New Class called "ImageUpload" in Parse
    ParseObject imgupload = new ParseObject("UserPhotos");

    // Create a column named "ImageName" and set the string
    imgupload.put("photoName", newTitle);

    // Create a column named "ImageFile" and insert the image
    imgupload.put("photo", file);

    // Create the class and the columns
    imgupload.saveInBackground(new SaveCallback() {

        @Override
        public void done(ParseException parseException) {
            if (parseException == null) {
                uploadedCallback.onUpdated(true, file.getUrl());
            } else {
                uploadedCallback.onUpdated(false, "");
            }
        }
    });
}
```

5.3.3. Tecnologías y bibliotecas utilizadas en la App.

En este punto procedemos a detallar sobre las diferentes tecnologías que se utilizan en Android, las diferentes bibliotecas de terceros que hemos utilizado, ya sea por su potencia, o por su utilidad como capa intermedia en el manejo del código.

Se ha tenido cuidado en escoger siempre bibliotecas que contengan código libre, no incumpliendo ninguna normativa de licencias que pudiese provocar un problema legal.

5.3.3.1. Persistencia de Datos- SQLITE – ADA FRAMEWORK

Android, para la persistencia de datos utiliza internamente SQLITE. SQLite es un motor de bases de datos muy popular en la actualidad por ofrecer un tamaño reducido, no hacer uso de servidor, una configuración sencilla, transaccional y por supuesto de código libre.

No obstante, pese a tener todas estas características la creación de la base de datos finalmente suele ser engorroso. En Android la forma común de trabajo con una base de datos es:

- 1) Crear la clase java con el modelo .
- 2) Definir una clase NO instanciable con el esquema de la Base de datos.
- 3) Para la conexión a la Base de datos será mediante la utilización de una clase auxiliar de tipo SQLiteOpenHelper que modificaremos a nuestros requerimientos.
- 4) Finalmente habrá que crear la clase DAO con las queries, utilizando una sintaxis compleja y llamando a los nombres de los campos y tablas de una forma laboriosa y poco optimizada, generando un código poco legible.

Para evitar tener que realizar estos pasos hemos utilizado un ORM, una biblioteca de persistencia llamada ADA FRAMEWORK ^{61, 62}. Este ORM nos permite despreocuparnos de lo relacionado con la base de datos. No será necesario el paso de crear las sentencias SQL de creación de tablas o todas las clases DAO del modelo.

El funcionamiento es muy simple, debemos crear nuestras clases de entidad y mediante anotaciones, estableceremos sus propiedades y relaciones.

Ada Framework también posee un databinder que maneja la relación entre nuestros Views y los atributos de las entidades (), permite llenar Listados de una forma auto-gestionada, etc.

En este ORM cabe destacar su potencia. Si lo comparamos con ORMLite, otro ORM muy extendido en la comunidad, la velocidad de ejecución es muy superior. Es importante su pequeño tamaño, apenas 72 kb. Un tamaño muy ajustado que nos ayuda a mantener comedido el tamaño de nuestra App.

El uso de Ada nos facilita enormemente el código y su entendimiento, pues nos libera de las clases engorrosas de SQL y nos permite crear toda la base de datos en dos sencillos pasos:

- 1) Mediante anotaciones, definir los tipos sobre las clases del modelo
- 2) Crearnos un “Manager” o DAO que nos permita ejecutar las operaciones que programemos, la sintaxis de este será propia del Framework pero es sencilla y clara.

Para el primer paso deberemos extender nuestras clases Java del modelo de la clase ENTITY, la cual le indica al framework que ese modelo generará una tabla, mediante anotaciones especificar el nombre de la tabla (@Table) y en cada uno de los campos que contenga la clase, especificar el tipo que generará en el SQLite y un nombre.

```
3 import com.mobandme.ada.Entity;
4 import com.mobandme.ada.annotations.Table;
5 import com.mobandme.ada.annotations.TableField;
6
7 @Table(name = "tWord")
8 public class Word extends Entity {
9
10     @TableField(name = "name", datatype = DATATYPE_STRING, unique = true)
11     public String name;
12
13     @TableField(name = "level", datatype = DATATYPE_INTEGER)
14     public int level;
15
16     @TableField(name = "definition", datatype = DATATYPE_STRING)
17     public String definition;
18
19     @TableField(name = "language", datatype = DATATYPE_INTEGER)
20     public int language;
21 }
```

Observamos en el código como especificamos nuestra tabla “tWord” y cada uno de los campos, las columnas que contendrá el SQL especificamos el tipo de dato. ADA, contiene los siguientes campos compatibles:

DATATYPE_BOOLEAN.*	DATATYPE_INTEGER.
DATATYPE_LONG.	DATATYPE_DOUBLE.
DATATYPE_REAL.	DATATYPE_TEXT.
DATATYPE_STRING.	DATATYPE_DATE.
DATATYPE_BLOB.	DATATYPE_ENTITY.
DATATYPE_ENTITY_LINK.	

** En SQLite no existe el tipo de datos booleanos así que se le trata internamente como un integer con el valor 1-0.*

Una vez tenemos definidas todos nuestros modelos, que se corresponderán a las tablas, hemos elaborado una clase denominada “Manager” en la cual, para que cumpla con los requisitos de Ada, debe heredar de la clase Object-Context. Aunque no es necesario, ni lo especifica la documentación de ADA, he considerado óptimo crear un singleton con la clase de acceso a la base de datos, siendo accesible desde cualquier clase, pues va a estar instanciada en la clase de “MyAplication” asociada a la actividad principal de la aplicación.

Dentro de esta clase Manager, existen “Objectsets”, que se corresponderán con los datos en memoria de los objetos que hagamos en las diferentes queries a base de datos. Esta característica de mantener los datos en memoria, hace que el número de accesos a la base de datos se reduzca enormemente y la velocidad de consulta sea mucho mayor, como contrapartida tiene que la aplicación puede tener un mayor consumo de memoria, pero tratándose de una índole de datos de tipo String, la cantidad de datos que tendríamos que generar es tal, que ni se contempla como posible.

```

35 public class Manager extends ObjectContext {
36
37     public ObjectSet<MyUser> user;
38     public ObjectSet<Word> Questions;
39     private static Manager mInstance = null;
40     private Context context;
41
42     // Singleton
43     public static Manager defaultManager(Context context) {
44         if (mInstance == null) {
45             mInstance = new Manager(context);
46         }
47         return mInstance;
48     }
49
50     private Manager(Context context) {
51         super(context);
52
53         this.context = context;
54
55         try {
56             this.user = new ObjectSet<MyUser>(MyUser.class, this);
57             this.Questions = new ObjectSet<Word>(Word.class, this);
58         } catch (AdaFrameworkException e) {
59             e.printStackTrace();
60         }
61     }
62 }

```

En la ilustración se observan la declaración de los ObjectSet, la instanciación del singleton y el constructor por defecto de la clase.

Una vez creado el Manager y la lógica de creación, simplemente nos queda crear cada uno de los métodos de gestión de la base de datos.

```

public boolean insertUser(MyUser mUser) {
    try {
        mUser.setStatus(Entity.STATUS_NEW);
        user.add(mUser);
        user.save();
        return true;
    } catch (AdaFrameworkException e) {
        e.printStackTrace();
        return false;
    }
}

```

Hay que detallar, que cada vez que se agrega un objeto se debe definir si es nuevo (STATUS_NEW), una actualización (STATUS_UPDATED) o de borrado (STATUS_DELETED).

5.3.3.2. Carga de imágenes, Universal Image Loader.



Aunque no hay numerosos recursos gráficos en la aplicación, considero una buena práctica el uso de esta librería SIEMPRE que se use cualquier imagen que se vaya a descargar de internet.

Esta potente librería también de código abierto hace una gestión eficaz de la memoria, evitando los temidos “memory override” que tanto se dan en Android, además este tipo de fallo depende mucho del dispositivo, por lo que es muy probable que en nuestra batería de dispositivos se ejecute correctamente y a la hora de l despliegue haya ciertos terminales donde falle. Haciendo uso de este gestor de imágenes, reducimos al mínimo esta posibilidad.

A su vez, esta librería hace un pre-cacheo de las imágenes, con lo que guarda cacheada una copia de la imagen, previniendo que cada vez que se acceda a el recurso haya que esperar a la conexión, ofreciendo mejoras muy substanciales de velocidad y de consumo de datos.

El uso de la librería es muy simple, declararemos una instancia que será un singleton a la cual accederemos desde los diferentes puntos de la aplicación para realizar la carga de imágenes.

1- Declaración en MyApplication:

```
// UNIVERSAL IMAGE LOADER SETUP
ImageLoaderConfiguration config = new ImageLoaderConfiguration.Builder(
    getApplicationContext())
    .defaultDisplayImageOptions(
        (new DisplayImageOptions.Builder())
            .bitmapConfig(Bitmap.Config.RGB_565)
            .imageScaleType(
                ImageScaleType.IN_SAMPLE_POWER_OF_2)
            .cacheOnDisc(true).cacheInMemory(true)
            .considerExifParams(true).build())
    .threadPoolSize(5)
    .denyCacheImageMultipleSizesInMemory()
    .discCache(
        new UnlimitedDiscCache(StorageUtils
            .getCacheDirectory(this)))
    .memoryCache(new FIFOLimitedMemoryCache(4096)).build();
ImageLoader.getInstance().init(config);
// END - UNIVERSAL IMAGE LOADER SETUP
```

En este caso he optado por aplicar algunas restricciones a la configuración por defecto, como el prever que se guarden varios tamaños, y forzar siempre el cacheado.

2.- Obtener el objeto image loader en las clases donde lo vayamos a usar:

Para recuperar el objeto, se ha considerado como buena praxis, en vez de estar llamándolo continuamente, comprobando el singleton etc. (lo hace internamente la librería), en los casos donde se vaya a utilizar se ha declarado una variable a la cual llamare en la carga de las imágenes.

```
private ImageLoader imageLoader = ImageLoader.getInstance();
```

y posteriormente en cada imagen se hace la carga con la librería, pasándole la Url y el ImageView donde se cargará la imagen.

```
imageLoader.displayImage(mUser.profileImageUrl, imgAvatar);
```

Cabe destacar también que la librería posee las herramientas necesarias para cada vez que se obtiene un recurso y este va a ser pintado en pantalla, se comprueba que el ImageView destino sigue estando visible, pues en caso de intentar cargarlo en un recurso que no estuviese activo, Android generaría una excepción y pararía la aplicación.

La librería de Image Loader la podemos encontrar en:

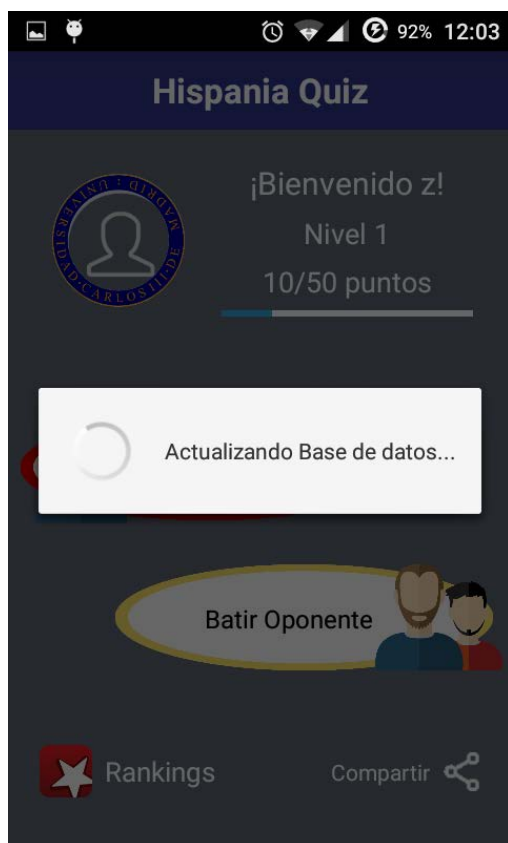
<https://github.com/nostra13/Android-Universal-Image-Loader>

5.3.3.3. Gestionando la red.

Esta aplicación hace uso intensivo de conectividad con los servidores backend, podríamos definir dos escenarios típicos con diferentes grados de uso de la red.

ESCENARIO SINGLE-PLAYER:

Para que un solo jugador pueda jugar, la parte más crítica en cuanto a conectividad es únicamente la primera vez que el jugador cree su cuenta de usuario, en este momento la aplicación comprobará si existe la Base de datos y en que versión se encuentra, como no encontrará el número de versión en SharedPreferences procederá a su descarga de Parse.com. Mientras se realiza esta descarga mantendremos la interfaz de usuario bloqueada pues , aunque en raras ocasiones, se pueden dar momento en el que mientras se sigan guardando datos en la Base de datos el usuario intentara jugar, esto provocaría un error de acceso a la base de datos, es una limitación de Android, si la Base de datos esta bloqueada en modo escritura, no podemos acceder a ella desde otro hilo, lanzará una excepción y bloqueará la aplicación.



Para prevenir esta situación se opta por mostrar un dialogo de descarga informando al usuario de la operación, además este dialogo le bloqueará solamente la primera ejecución o en su defecto si creamos una nueva versión de la Base de datos, situación que al ser distinta la versión online a la versión local, desecharía la versión local y recrearía la base de datos con la nueva información.

Ilustración 88 Detalle Dialogo Espera

ESCENARIO MULTIJUGADOR:

Para el modo multijugador se han implementado multitud de pequeños bloqueos para el usuario, que nos aseguran el correcto alineamiento de los datos entre el servidor y la App. Estos son necesarios pues si un usuario gana o pierde una partida deberemos esperar y resincronizar los listados de los turnos, también se ha agregado un actualizar automático cada 60 segundos por si mientras estamos en la pantalla de las listas de partidas el contrincante juega, este se nos actualice en vivo.

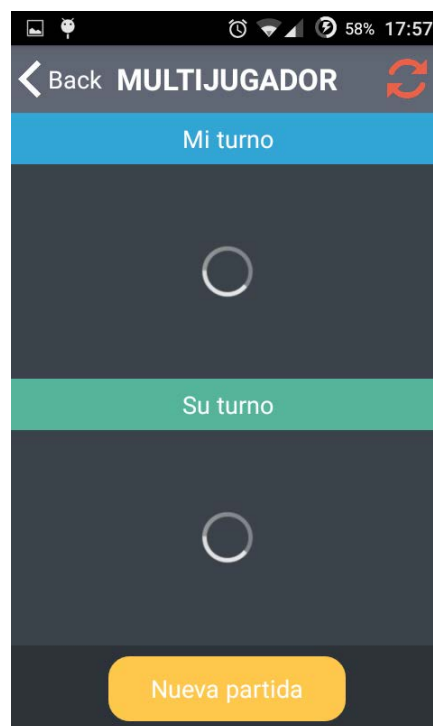


Ilustración 89 Detalle bloqueo interfaz multijugador

A parte de estos dos escenarios, todas las conexiones con cualquier servidor se han realizado en hilos secundarios y manejado las llamadas mediante Callbacks, en todos los casos se estudia si la respuesta es correcta o se ha producido un error realizando en cada caso las acciones que se correspondan, se adjunta una imagen con un método donde se aprecia claramente esta buena praxis.

```
public void getTopScores(final App42CallBack callBack) {
    if (callBack == null) {
        Log.e(App42Manager.class.getSimpleName(),
            "Callback cannot be null!");
    } else {
        scoreBoardService.getTopRankings(Constants.GAME_NAME,
            new App42CallBack() {
                @Override
                public void onSuccess(final Object response) {
                    new Handler(Looper.getMainLooper())
                        .post(new Runnable() {
                            @Override
                            public void run() {
                                callBack.onSuccess(response);
                            }
                        });
                }
                @Override
                public void onException(final Exception ex) {
                    new Handler(Looper.getMainLooper())
                        .post(new Runnable() {
                            @Override
                            public void run() {
                                callBack.onException(ex);
                            }
                        });
                }
            });
    }
}
```

5.3.3.4. Notificaciones PUSH

Una aplicación basada en el juego por turnos, es fundamental incorporar notificaciones Push que avisen al usuario cuando:

- ✓ Otro usuario le reta
- ✓ Le toca jugar por turno
- ✓ Ha ganado/ perdido una partida.

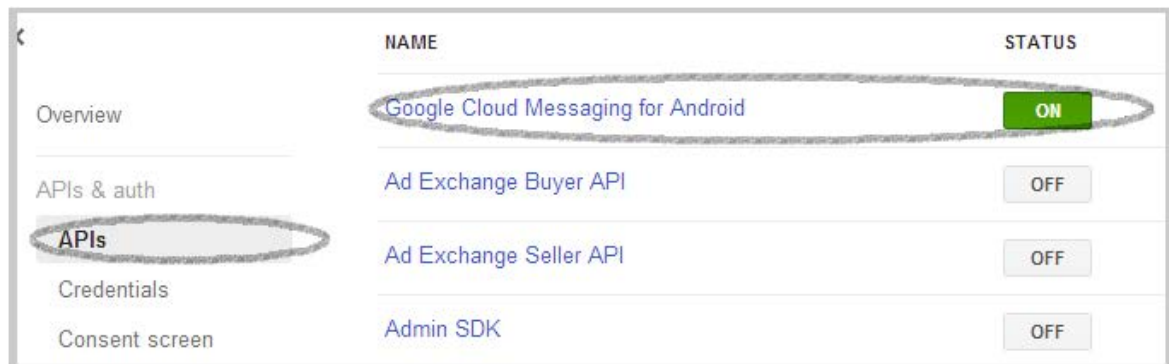
Para realizar las notificaciones Push usando APP42 necesitamos completar una serie de pasos:

Lado Servidor:

El primer requisito es haber creado la aplicación en la consola de Google, podemos encontrar los datos de acceso en el Anexo III de este documento.

Consola: <https://console.developers.google.com/start>

Una vez en ella deberemos Habilitar la Api de Google de “Google Cloud Messaging for Android”. Este paso habilitara que google identifique cada dispositivo de forma única.



**Ilustración 90 Detalle
Activación GCM I**

Y obtener el identificadnos único que se le da a nuestra aplicación para hacer las notificaciones PUSH.

Acceso a API pública

El uso de esta clave no requiere consentimiento ni acciones por parte del usuario ni concede acceso a la información de la cuenta. Además, no se utiliza para la autorización.

[Más información](#)

[Crear clave nueva](#)

Clave para las aplicaciones de servidor

Clave de la API	AlzaSyAcJOTVQM43LK15JA6H9Lcqb0udmQT3FR0
IPs	Se permite cualquier IP
Fecha de activación	28 abr. 2015 18:43:00
Activado por	pfcrafaelgonzalez@gmail.com (tú)

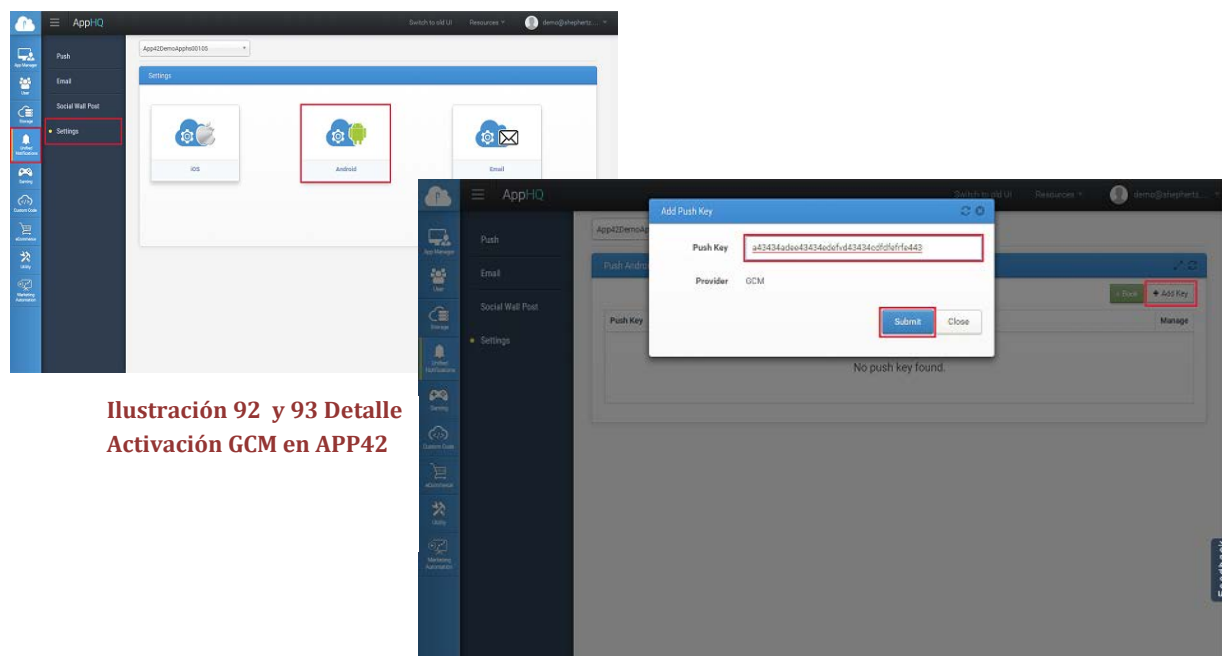
[Editar IP permitidas](#)

[Volver a generar la clave](#)

[Eliminar](#)

**Ilustración 91 Detalle
Activación GCM II**

Una vez en posesión de nuestro identificador, deberemos configurar el servidor de APP42 para que sepa cual es nuestro identificador para la plataforma Android.



**Ilustración 92 y 93 Detalle
Activación GCM en APP42**

Lado Cliente.

Para poder **recibir** notificaciones lo primero que tenemos que hacer es registrar el dispositivo que estamos utilizando con Google, para que nos facilite un identificador único, este identificador lo guardaremos en Shared preferences y así podremos usarlo siempre, y cuando hagamos un logout del usuario podremos dar de baja el dispositivo de las notificaciones.

```
public void run() {
    GoogleCloudMessaging gcm = GoogleCloudMessaging
        .getInstance(MyApplication.getContext());
    final String registrationId;
    try {
        registrationId = gcm.register(GCM_PUSH_SERVER_ID);
        SharedPreferencesManager defaultManager(context)
            .setGcmPushId(registrationId);
    }
```

Posteriormente deberemos registrar en nuestro backend el id que nos ha devuelto google, el id de Google cloud Messaging estamos atacando, este es el generado en el paso anterior, y el nombre de usuario a que se corresponde para así posteriormente permitir que app42 nos envíe las notificaciones correspondientes a el usuario que se desee.

```
pushNotificationService.storeDeviceToken(username,
    registrationId, new App42CallBack() {
```

El tercer paso tenemos que generar el código necesario en la aplicación, para ello vamos a crearnos una clase que llamaremos GCMReceiver en la que incorporaremos la lógica de las notificaciones. Esta clase extenderá de “wakefulBroadcastReceiver”, que se caracteriza por poder recibir las notificaciones aunque el dispositivo este en estado de suspensión y en caso de asociarle un servicio, nos asegurará que el dispositivo se mantendrá activo hasta la terminación del servicio. En esta clase sobrescribiremos el método “onReceive” que será el llamado cada vez que llegue una nueva notificación.

En el caso que nos atañe, no vamos a utilizar el servicio, pues el cometido es solamente mostrar la notificación, a la cual le asociaremos un Intent para que si el usuario la presiona nos habrá la App en la ventana inicial.

Para que el sistema operativo sepa que nuestra aplicación puede recibir notificaciones Push, en el manifest le deberemos indicar que tipo de notificación esperamos.

```
<receiver
    android:name="com.uc3m.hispania.utils.gcm.GCMReceiver"
    android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
    </intent-filter>
    <category android:name="com.uc3m.hispania" />
</receiver>
```


Una vez que el S.O. recibe la notificación nos la pasa a nuestra app, y nosotros a la clase GCMReceiver, en donde se llamará al método onReceive, en él para evitar posibles notificaciones malformadas las filtraremos aseguramos que van a salir completas.

```
public class GCMReceiver extends WakefulBroadcastReceiver {

    public static final int NOTIFICATION_ID = 1;

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle bundle = intent.getExtras();
        Log.d("", "Bundle: " + bundle);
        Log.d("GCMReceiver", "Push recibido");
        try {
            if (bundle.getString("alert")!=null) {
                JSONObject jsonObject = new JSONObject(bundle.getString("alert"));
                sendNotification(context, jsonObject.get("alert").toString());
            }

        } catch (JSONException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

Una vez que hemos recibido la notificación y verificamos que esta completa, pasamos a agregarla a la barra de notificaciones. Le especificamos el título, la imagen que va a tener y el texto que nos recibido en la notificación detallándonos el cometido.

```
private void sendNotification(Context context, String msg) {
    NotificationManager mNotificationManager = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);

    PendingIntent contentIntent = PendingIntent.getActivity(context, 0,
        new Intent(context, MainActivity.class), 0);

    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(
        context).setSmallIcon(R.drawable.hispania_200)
        .setContentTitle("Hispania Quiz")
        .setStyle(new NotificationCompat.BigTextStyle().bigText(msg))
        .setContentText(msg);
    mBuilder.setAutoCancel(true);

    mBuilder.setContentIntent(contentIntent);
    mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
}
```

Para **enviar** notificaciones desde el dispositivo, App42 nos ofrece un método muy sencillo de utilizar, el cual esta implementado en el App42Manager.

```
pushNotificationService.sendPushMessageToUser(
    username.toLowerCase(), json.toString(),
    new App42CallBack() {
```

5.3.3.5. Google Analytics.

Google Analytics ^{64, 65, 66} es una herramienta que nos da Google de forma gratuita a los desarrolladores para recabar información relevante sobre la aplicación o web. Desde el punto de vista que nos interesa la Aplicación móvil nos da información muy útil tanto desde un nivel de mercado hasta un nivel de mantenimiento de la aplicación y corrección de bugs.



- Desde un Punto de vista **comercial** Google Analytics nos permite ver en que regiones se descarga nuestra Aplicación, el público sobre el que causa más impacto. También nos ofrece información de cuando han sido las instalaciones, el tiempo medio que perdura el usuario en la aplicación, que pantallas visita o cuales entra menos o de todas las instalaciones cuantos usuarios permanecen Activos actualmente.
- Desde un punto de vista de **desarrollo**, Google Analytics nos da información relevante sobre donde se han producido los diferentes bloqueos en nuestra aplicación, permitiéndonos identificar

Los principales apartados con los que cuenta la herramienta son:

- **Informes de análisis de adquisiciones y usuarios:** es la pantalla principal en donde de un solo vistazo tendremos acceso a toda la información de nuevos usuarios, usuarios recurrentes, procedencia o características de los usuarios. También podemos ver como interactúan los usuarios con la las diferentes partes de la aplicación.

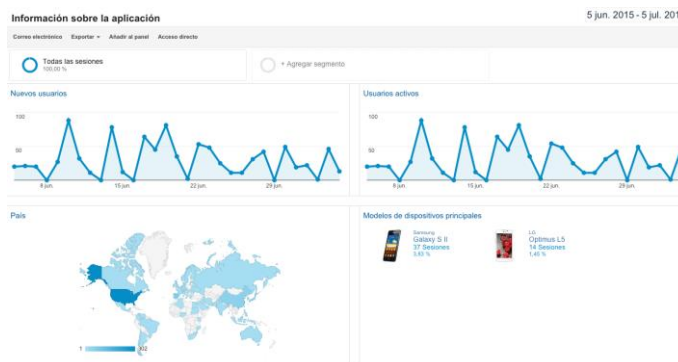


Ilustración 94 Captura Pantalla Google Analytics

- **Tiempo Real:** Nos ofrece el poder ver en cada instante las diferentes acciones que realiza el usuario, mandando los ítems que hayamos marcado en el código para hacer un seguimiento. También nos permite ver en tiempo real donde se esta ejecutando nuestra app, que pantallas son las más visitadas etc.

- **Audiencia:** Nos ofrece información más específica de los usuarios, el tiempo medio de sesión, idioma de los terminales, proveedor de servicios y un amplísimo etc.

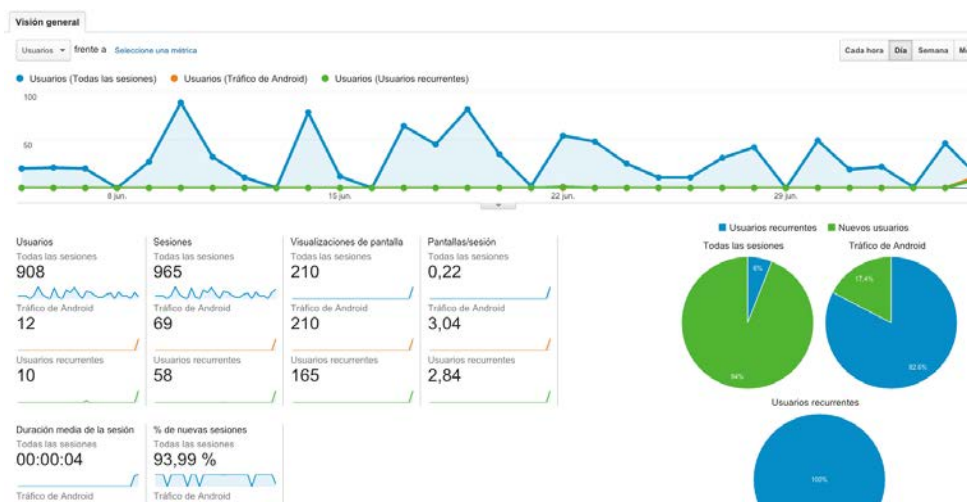


Ilustración 95 Captura Pantalla Google Analytics- Audiencia

Comportamiento: Nos ofrece información sobre los diferentes comportamientos no deseados que tenga nuestra App, bloqueos y nos dice exactamente en que línea de código se ha producido una excepción. Muy útil pues se identifican de manera sencilla incompatibilidades con dispositivos o fallos de programación.

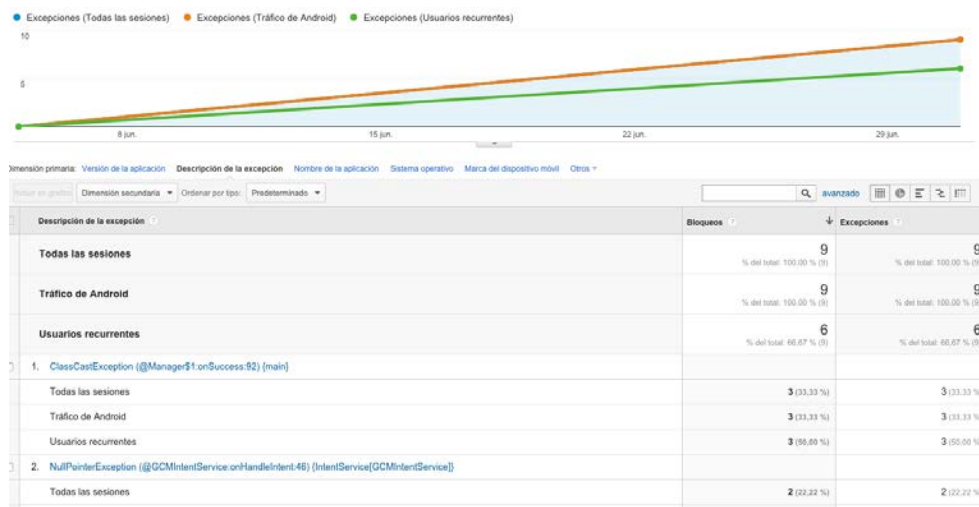


Ilustración 96 Captura Pantalla Google Analytics- Comportamiento

5. `NullPointerException (@ParseManager:downloadNewQuestionsWithSkip:168) {Thread-2183}`

Apreciamos en detalle como marca la excepción y la línea donde se produce:



Google AdMob, es una red publicitaria para móviles que fue comprada por google, es independiente de la red de anuncios para móviles google y nos permite, tanto invertir dinero en para promocionar nuestra app, o lo que queramos publicitar o como ganar dinero cediendo espacio en nuestra aplicación para publicidad de terceros.

AdMob Necesita para funcionar de una cuenta que tenga asociada una serie de datos de índole personal por lo tanto y debido a la simplicidad del cambio, pues solo se ha de cambiar el identificador de AdMob en la aplicación, he optado por usar **una cuenta personal para el gestor de anuncios**.

Para integrar Google AdMob debemos completar los siguientes hitos:

- Tener una Aplicación Android en la que queramos incluir la publicidad, Dicha App debe tener como SDK mínimo la API-13.
- Registrarnos en Google AdMob.
- Una vez tengamos Cuenta, deberemos crear nuestra aplicación y seleccionar el tipo de espacio que vamos a dejar a terceros para que se publiciten, pudiendo ser Banners de diferentes tamaños, o como es nuestro caso una pantalla completa, lo cual recibe el nombre de “interstitial”, esto nos facilitará un id que será el que pongamos en nuestra App y el que nos identifique para realizar nuestra App, tanto para enviar anuncios a ella o recibir los de terceros y cobrar por ello.
- Deberemos incluir la librería del SDK de AdMob en nuestro proyecto
- Para integrar AdMob en nuestro código hemos optado por crear una clase propia que sea la encargada de gestionar la actividad del interstitial de publicidad. Esta clase la hemos hecho heredar de

FragmentActivity, pudiendo utilizar el interstitial como un fragment, necesario debido a nuestra lógica d aplicación y el uso de viewPagers, y se cumple con la necesidad del SDK de AdMob de pertenecer a una actividad. Esta deberá ser incluida en el Manifest.

```
public class InterstitialActivity extends FragmentActivity {
    private static String AD_UNIT_ID = "ca-app-pub-6044385826066685/2352057257";
    private InterstitialAd interstitialAd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Crear adView.
        interstitialAd = new InterstitialAd(this);
        interstitialAd.setAdUnitId(AD_UNIT_ID);

        // Iniciar una solicitud genérica.
        AdRequest adRequest = new AdRequest.Builder().addTestDevice(
            "A0A7A032AFBFF29142A8A6C729D9569F").build();

        // Cargar adView con la solicitud de anuncio.
        interstitialAd.loadAd(adRequest);
    }

    public void displayInterstitial(final AdListener listener) {
        interstitialAd.setAdListener(listener);
        if (interstitialAd.isLoaded()) {
            interstitialAd.show();
        } else {
            new Handler().postDelayed(new Runnable() {
                @Override
                public void run() {
                    displayInterstitial(listener);
                }
            }, 200);
        }
    }
}
```

Al tratarse se un interstitial a pantalla completa no deberemos definir una zona en nuestros ficheros XML donde se mostrarán los anuncios, lo cual nos simplifica la integración de AdMob, los anuncios más grandes nos permiten maximizar nuestro beneficio y no nos aumentará la complejidad del diseño de las pantallas al tener que contar con una zona para anuncios.

Google AdMob aumentará una pequeña cantidad de saldo por cada vez que se muestre un anuncio en nuestra aplicación. A su vez, si dispusiésemos de varias aplicaciones podríamos establecer reglas para hacer promoción interna entre nuestras diferentes Apps.

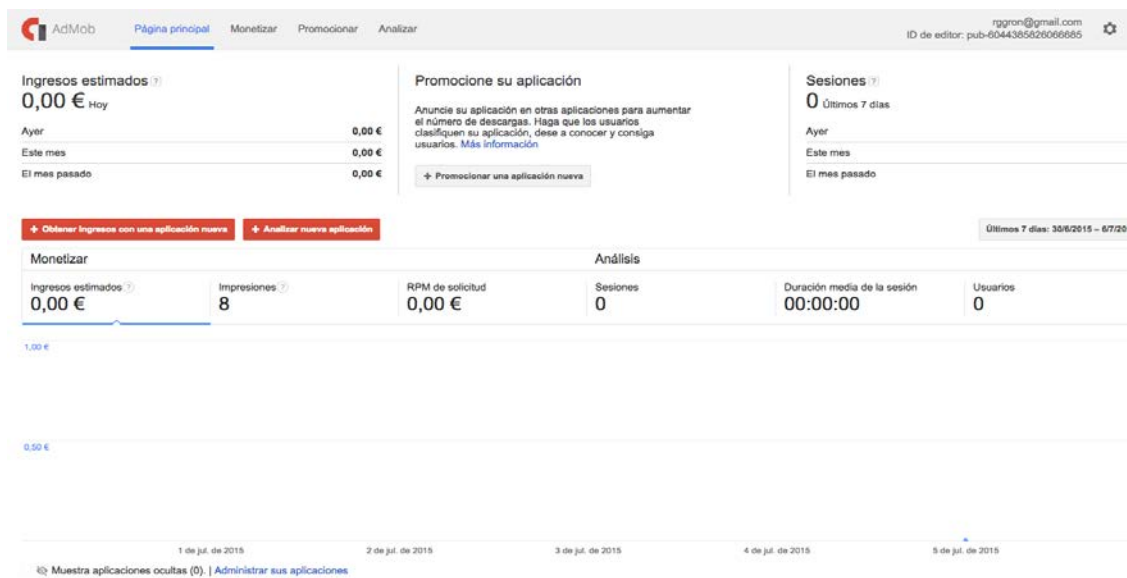


Ilustración 97 Captura Pantalla Google AdMob

En la pantalla inicial de AdMob podemos ver el saldo que llevamos hasta la fecha, obviamente en el caso que nos acata es 0.0€ (lamentablemente).

A su vez, la forma de monetizar con AdMob cambia según el espacio que cedamos a la publicidad y lo efectiva que esta sea. A mayor tamaño de Pantalla más nos pagarán por la publicidad, a su vez, si el usuario final hace clic sobre el banner / interstitial se considerará como anuncio visto y el pago será muy superior. No obstante, para hacernos una idea, los pagos esta en torno a 1€ por cada 1000 visualizaciones.

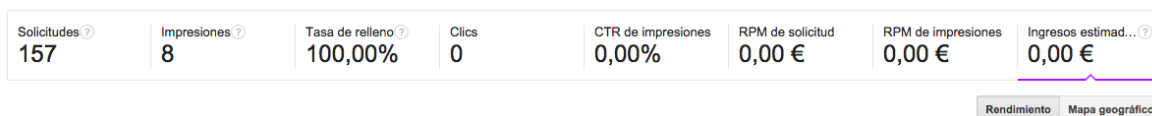


Ilustración 98 Captura Pantalla ingresos AdMob

Se puede apreciar como se lleva un control absoluto de las diferentes variables, impresiones, clicks realizados, acceso a datos geográficos y demográficos, porcentaje de éxito de clic etc.

CAPÍTULO 6

REDES SOCIALES Y GAMIFICACIÓN

6. GAMIFICACIÓN, REDES SOCIALES Y SHARING

En este capítulo se detallarán todos los pasos y etapas que conforman el proceso de sharing de la app, Gracias al cual se intentará conseguir una rápida expansión de la aplicación.

6.1. Integración con Facebook.

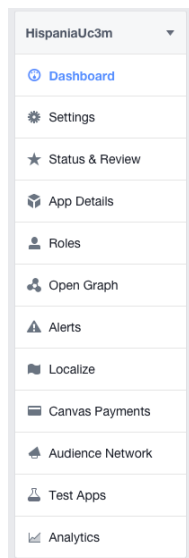


Debido a la fuerte proliferación de las redes sociales y en especial la abrumadora irrupción de Facebook en nuestra vida diaria, lo convierten en una herramienta indispensable, ya no solo para nuestro objetivo de gamificación, si no que es un requisito de indispensable cumplimiento en cualquier desarrollo que realicemos.

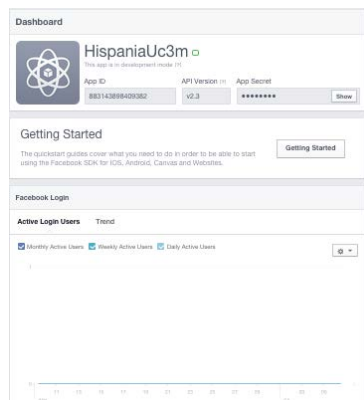
Utilizar la consola de Facebook y su SDK en general es uno de los mayores quebraderos de cabeza a los que se enfrenta un programador, la documentación que aporta Facebook es imprecisa, muy mal organizada y en donde encontrar, en caso de que nos ocurra, un posible fallo en nuestro código es una pérdida absoluta de años de vida. No existe una documentación que este bien organizada y este actualizada. Los Permisos sobre políticas de privacidad y modo de funcionamiento pueden cambiar con cada SDK de Facebook, para cada reléase de SDK Facebook se compromete a mantenerla durante dos años, pero esta situación genera un conflicto en la documentación y en los problemas de los programadores que se exasperan buscando su error. No hay nada como “pasear” por los foros para darse cuenta de esta situación. Problemas de tipo “pero si esto lo hago en no se cuantas aplicaciones, siendo usuario, donde esta como hacerlo?” y no encontrar respuesta o descubrir en un sitio remoto o gracias a algún foro que esa funcionalidad ha sido eliminada o capada. Hacer una petición, bajada directamente del ejemplo que pone Facebook y recibir un “error desconocido” son problemas que se encuentran a la orden del día para cualquier persona que tiene la osadía de enfrentarse a hacer una integración con Facebook.

Para integrar Facebook en nuestra aplicación el primer paso que debemos realizar es registrarse como desarrollador en Facebook Developers (se adjunta datos de cuenta [en adjunto X](#)) en donde Registraremos nuestra App para conseguir nuestro identificador único sobre el que identificará nuestra App.

6.1.1. La consola de FB

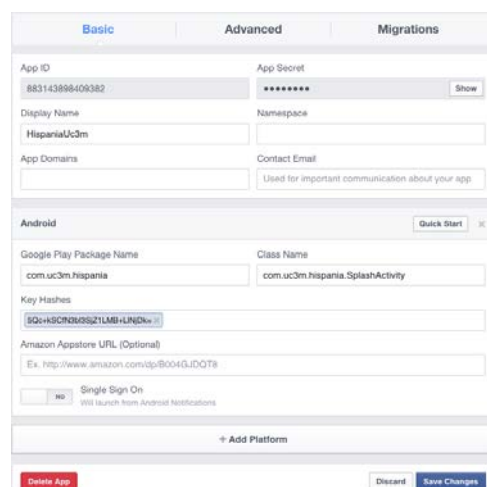


La consola de Facebook contiene multitud de elementos que hay que configurar antes de que podamos, por ejemplo compartir nuestra aplicación por Facebook, vamos a detallar las principales características de cada elemento.

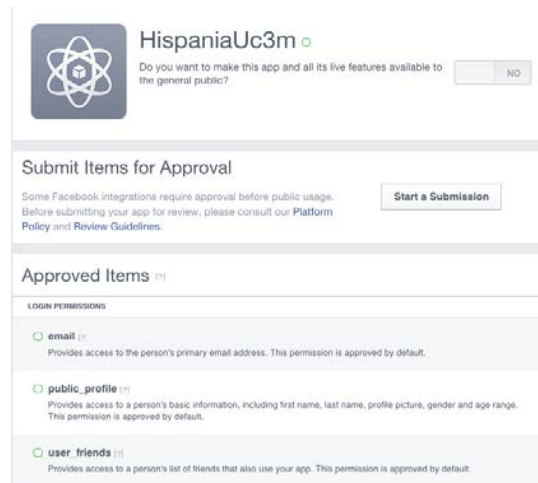



SETTINGS: Información específica para la integración y configuración de la app con Facebook. El id, el nombre del paquete, la clase de inicio, y los keyhashes de los móviles de desarrollo, pues si no dará un error al intentar probarla. Todos estos parámetros deberemos haberlos rellenado antes de poder usar Facebook, pues en caso de faltar alguno no nos identificará correctamente la App.

DASHBOARD: el Dashboard nos ofrece la información general de la aplicación, el número de instalaciones, el numero de sesiones activas de FB, además de mostrarnos también un resumen del nº de llamadas hechas a la Api, o los errores generados.



STATUS & REVIEW: Desde finales del 2014 aproximadamente, toda aplicación que



HispaniaUc3m 

Do you want to make this app and all its live features available to the general public? ☐ NO

Submit Items for Approval

Some Facebook integrations require approval before public usage. Before submitting your app for review, please consult our [Platform Policy](#) and [Review Guidelines](#). [Start a Submission](#)

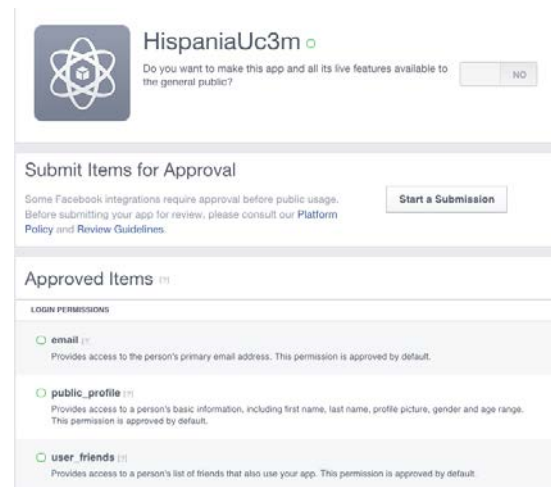
Approved Items ⁽¹⁾


LOGIN PERMISSIONS

- ☒ **email** ⁽¹⁾
Provides access to the person's primary email address. This permission is approved by default.
- ☒ **public_profile** ⁽¹⁾
Provides access to a person's basic information, including first name, last name, profile picture, gender and age range. This permission is approved by default.
- ☒ **user_friends** ⁽¹⁾
Provides access to a person's list of friends that also use your app. This permission is approved by default.

vayamos a integrar con Facebook debe estar validada por ellos, hay que subirles una guía de uso y manejo especificando con ejemplos MUY CLAROS que permisos y en que partes de nuestra app se hacen las diferentes llamadas a su sistema y con que finalidad. Es muy común encontrarse con comentarios del personal a cargo como “eso no es buena practica” o sencillamente un “no lo encuentro”, con lo que la guía que les enviemos ha de ser lo más detallada posible, incorporar pantallazos...

APP DETAILS: En este apartado deberemos detallar toda la información de la aplicación, nombre descripción, que permisos se piden y para qué, todo ello de cara a el usuario, es la visión que va atener un usuario de nuestra App.



HispaniaUc3m 

Do you want to make this app and all its live features available to the general public? ☐ NO

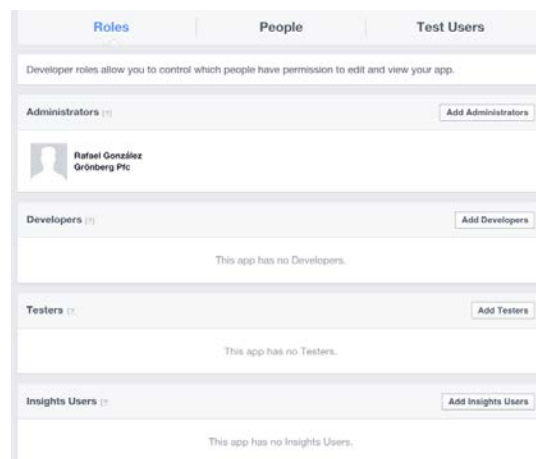
Submit Items for Approval

Some Facebook integrations require approval before public usage. Before submitting your app for review, please consult our [Platform Policy](#) and [Review Guidelines](#). [Start a Submission](#)

Approved Items ⁽¹⁾

LOGIN PERMISSIONS


- ☒ **email** ⁽¹⁾
Provides access to the person's primary email address. This permission is approved by default.
- ☒ **public_profile** ⁽¹⁾
Provides access to a person's basic information, including first name, last name, profile picture, gender and age range. This permission is approved by default.
- ☒ **user_friends** ⁽¹⁾
Provides access to a person's list of friends that also use your app. This permission is approved by default.



Roles | **People** | **Test Users**

Developer roles allow you to control which people have permission to edit and view your app.

Administrators ⁽¹⁾ [Add Administrators](#)

 **Rafael González Grönberg Pfc.**

Developers ⁽¹⁾ [Add Developers](#)

This app has no Developers.

Testers ⁽¹⁾ [Add Testers](#)

This app has no Testers.

Insights Users ⁽¹⁾ [Add Insights Users](#)

This app has no Insights Users.

ROLES: En esta pantalla se especifican los usuarios que son Administradores de la App, los desarrolladores y los testers, se debe completar pues en su defecto nos será imposible realizar pruebas de integración pues nos rechazará por sistema cualquier usuario. Existe la posibilidad de crear test Users para no tener que estar utilizando cuentas reales.

Open Graph: Esta parte es bastante reciente, es la nueva vía que utiliza Facebook para poder compartir mensajes autogenerados , es decir, si el usuario edita el texto puede compartir lo que quiera, pero si nuestra app quisiese compartir algo, tendríamos que haber predefinido una historia previamente. Esto es, una serie de verbos con sus tiempos verbales, unos sujetos

y una serie de adjetivos, por ejemplo para compartir actualmente Facebook sólo nos deja hacer frases tipo “*JUAN* ha comido en *RESTAURANTE PACO* con *JULIANA*, en donde las cursivas son las únicas palabras que mandaríamos desde la app y las cuales tienen que estar previamente definidas y aprobadas por Facebook.

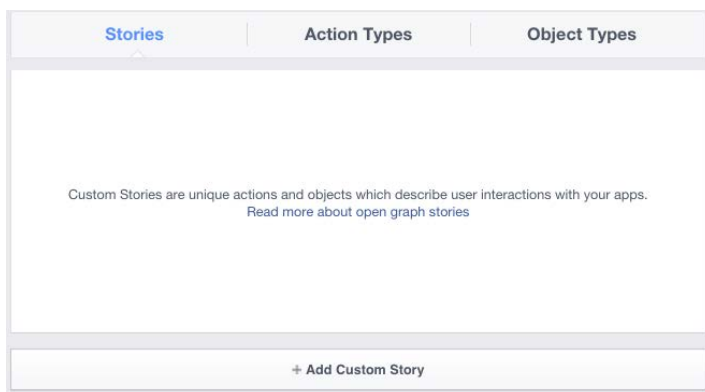
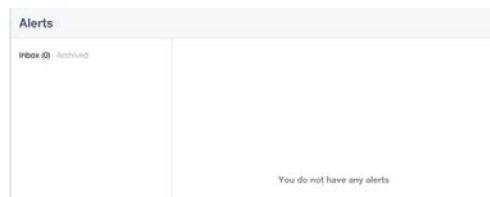


Ilustración 99 Tabla ilustrada de pantallas y funcionalidades Facebook



Alerts: cuando subamos la app a revisión con Facebook, aquí se nos indicarán los requisitos que no cumplimos.

Localize: En esta pestaña podemos seleccionar los diferentes idiomas en los que estará disponible nuestra app, y sus correspondientes traducciones.

Canvas Payments, Audicence, TestApps y Analitics: de forma similar a la ofrecida por google nos ofrece datos para poner publicidad, hacer campañas, ver estadísticas de usuarios...

6.1.2. Integrando el código de FB.

En este apartado, se hablará de los cambios acometidos sobre la aplicación para poder dotarla de las funcionalidades más básicas de conectividad con Facebook, se ha preferido optar por aumentar los medios de divulgación de la App antes que centrarnos en utilizar todas las funcionalidades de Facebook por lo que se ha optado por utilizar el medio por defecto para compartir en redes sociales. Este sistema nos permite comentar en Facebook nuestros propios comentarios con un link a nuestra App, pero en su defecto no nos permitirá realizar historias, es decir, no podremos mandar mensajes de tipo “Juan ha retado a lola a Hispania Uc3m”.

Esta decisión de diseño la he tomado porque a la hora de la implementación es mucho más costosa, además con la nueva política de privacidad de Facebook no se nos permite acceder a datos de ningún usuario si ese usuario no esta previamente registrado en nuestra App, con lo que crear historias se nos quedaría un poco cojo.

Inicialmente se quería poner una frase predefinida en el cuadro de compartir de Facebook, pero también en la última versión del SDK han avisado que eso es una practica que supondrá el rechazo directo de la aplicación.

Es decir, con las limitaciones para publicar existentes y previendo que el juego no va a llegar a un éxito masivo se ha optado por la utilización de compartir vía intent, que nos facilita enormemente el código permitiéndonos dedicar los recursos a otras tareas y no limita

en exceso la funcionalidad. No obstante, si la app previéramos un rápido crecimiento o descubriéramos una masiva aceptación, sería un requisito muy deseable implementar un Manager que trabajara completamente la sesión de Facebook y poder crear Historias de tipo “open graph”

```
public static boolean shareFacebook(String text, Context context) {
    Intent shareIntent = new Intent(Intent.ACTION_SEND);
    shareIntent.setType("text/plain");
    shareIntent.putExtra(Intent.EXTRA_TEXT, text);
    shareIntent.putExtra(Intent.EXTRA_SUBJECT, context
        .getResources().getString(R.string.app_name));

    List<ResolveInfo> activityList = context.getPackageManager()
        .queryIntentActivities(shareIntent, 0);
    for (final ResolveInfo app : activityList) {
        if ((app.activityInfo.name).contains("facebook")) {
            final ActivityInfo activity = app.activityInfo;
            final ComponentName name = new ComponentName(
                activity.applicationInfo.packageName, activity.name);
            shareIntent.addCategory(Intent.CATEGORY_LAUNCHER);
            shareIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
                | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
            shareIntent.setComponent(name);
            context.startActivity(Intent.createChooser(shareIntent,
                "Share by Facebook"));
            return true;
        }
    }
    return false;
}
```

6.2. Integración con Twitter.



social concreta poseerá la app asociada y si no hace uso, no la tendrá, pero entiendo que tampoco hace uso de ella con lo que no compartiría la app por ese medio.

De Twitter no vamos a utilizar el SDK pues no vamos a hacer una integración con Facebook para darnos de alta en la app, el único uso que le queremos dar es el poder promocionar la APP, con lo que el realizar un compartir vía Intent nos es suficiente.

Si quisiésemos poder utilizar la api de Twitter, que actualmente han incorporado fabric, un SDK propio pues hasta hace menos de 5 meses los SDK eran los desarrollados para java por la comunidad. Nuestro primer paso será darnos de Alta en Twitter Developers y crear una nueva aplicación, para conseguir nuestro Identificador para la App en Twitter.

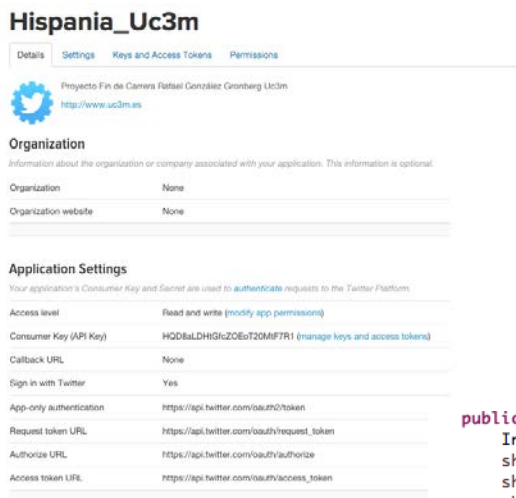


Ilustración 100 Captura pantalla configuración Twitter

Actualmente se hace una compartición mediante una publicación estándar en Twitter.

Por similar motivo que en Facebook se ha optado por una integración mediante Intent, es decir, solo se podrá utilizar este medio si el usuario tiene instalada la APP de Twitter en el dispositivo. Sobreentendemos que una mayoría de la gente que haga uso de una red

Con esta ya tendríamos los pasos necesarios en caso de querer mandar la app a amigos concretos o comentar sabiendo el nombre de otro usuario, ambas opciones se nos antojan más plausibles para una fase dos del proyecto.

```
public static boolean shareTwitter(String text, Context context) {
    Intent shareIntent = new Intent(Intent.ACTION_SEND);
    shareIntent.setType("text/plain");
    shareIntent.putExtra(Intent.EXTRA_TEXT, text);
    shareIntent.putExtra(Intent.EXTRA_SUBJECT, context
        .getResources().getString(R.string.app_name));

    List<ResolveInfo> activityList = context.getPackageManager()
        .queryIntentActivities(shareIntent, 0);
    for (final ResolveInfo app : activityList) {
        if ((app.activityInfo.name).contains("twitter")) {
            final ActivityInfo activity = app.activityInfo;
            final ComponentName name = new ComponentName(
                activity.applicationInfo.packageName, activity.name);
            shareIntent.addCategory(Intent.CATEGORY_LAUNCHER);
            shareIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
                | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
            shareIntent.setComponent(name);
            context.startActivity(shareIntent);
            return true;
        }
    }
    return false;
}
```

6.3. Integración con e-mail, SMS y WhatsApp.

Aprovechando la compartición vía Intent se han implementado también los métodos para poder compartir la App mediante SMS, e-mail y WhatsApp, esto esta enfocado sobre todo a la hora de no saber una respuesta, el intentar preguntarla de una manera rápida , con el tiempo tan limitado de 30 segundos para responder es difícil que le de tiempo a ver la palabra, pensarla, enviarla por WhatsApp y le respondan, pero si lo intenta, ya tenemos una compartición más de nuestra app.

```
public static boolean shareWhatsapp(String text, Context context) {
    Intent shareIntent = new Intent(Intent.ACTION_SEND);
    shareIntent.setType("text/plain");
    shareIntent.putExtra(Intent.EXTRA_TEXT, text);
    List<ResolveInfo> activityList = context.getPackageManager()
        .queryIntentActivities(shareIntent, 0);
    for (final ResolveInfo app : activityList) {
        if ((app.activityInfo.name).contains("com.whatsapp")) {
            final ActivityInfo activity = app.activityInfo;
            final ComponentName name = new ComponentName(
                activity.applicationInfo.packageName, activity.name);
            shareIntent.addCategory(Intent.CATEGORY_LAUNCHER);
            shareIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
                | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
            shareIntent.setComponent(name);
            context.startActivity(Intent.createChooser(shareIntent,
                "Share by Whatsapp"));
            return true;
        }
    }
    return false;
}
```



```
public static boolean shareEmail(String text, String subject, Context context){
    Intent emailIntent = new Intent(Intent.ACTION_SENDTO);
    emailIntent.setData(Uri.parse("mailto:"));
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
    emailIntent.putExtra(Intent.EXTRA_TEXT, text);

    try {
        context.startActivity(Intent.createChooser(emailIntent, "Send email using"));
        return true;
    } catch (android.content.ActivityNotFoundException ex) {
        return false;
    }
}
```

```
public static boolean shareSMS(String text, Context context) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        String defaultSmsPackageName = Telephony.Sms
            .getDefaultSmsPackage(context);
        Intent sendIntent = new Intent(Intent.ACTION_SEND);
        sendIntent.setType("text/plain");
        sendIntent.putExtra(Intent.EXTRA_TEXT, text);

        if (defaultSmsPackageName != null) {
            sendIntent.setPackage(defaultSmsPackageName);
        } else {
            return false;
        }
        context.startActivity(sendIntent);
        return true;
    } else // For early versions, do what worked for you before.
    {
        Intent sendIntent = new Intent(Intent.ACTION_VIEW);
        sendIntent.setData(Uri.parse("sms:"));
        sendIntent.putExtra("sms_body", text);
        context.startActivity(sendIntent);
        return true;
    }
}
```



CAPÍTULO 7

CONTROL DE CALIDAD

7. CONTROL DE CALIDAD

En este punto del proceso de desarrollo, nos preguntamos si todo el trabajo realizado hasta ahora con el prototipado en papel y el continuo preguntar a diferentes usuarios sobre sus opiniones y operativa para llegar a una funcionalidad, si realmente se ha sabido extraer ese conocimiento y plasmarlo un diseño acorde a las expectativas del proyecto.

El control de calidad en cuando a código y programación, robustez del aplicativo, ha sido constante mediante realización de pruebas unitarias en cada nueva funcionalidad que se ha implementado, tanto de la parte de nueva programación como de las funcionalidades previamente implementadas.

Habría sido interesante agregar al proyecto alguna herramienta de test unitarios, pero debido a la magnitud del proyecto se ha dado por buena la resolución manual de las pruebas en cada interacción, pues se considera que las pruebas son rápidas y el tiempo de generación de los test unitarios hubiese sido mayor.

7.1. Evaluación usabilidad

En lo referente a el triunfo de la aplicación, el principal punto que debemos estudiar es la usabilidad. El profundo conocimiento del programador sobre la aplicación puede dar por sentada muchas premisas y modelos de comportamiento que él ha interiorizado y adoptado como naturales y sencillas de la aplicación, ahora bien, expondremos nuestra beta de la aplicación a una serie de usuarios sobre los que el conocimiento de el aplicativo sea el estrictamente necesario y estudiaremos si con el diseño y los medios que les hemos puesto a su alcance son capaces de realizar con éxito una serie de pruebas que se establecerán en base a los diferentes casos de uso que se han propuesto anteriormente.

7.1.1. Perfiles evaluador.

Para la realización de esta prueba, y debido a la cercanía de estos, se ha decidido reclutar a 4 personas de mi entorno sobre los que no existe conocimiento previo de la aplicación.

ROL	Nº	CARACTERÍSTICAS PERFIL
Usuario experto	1	Desarrollador software con amplios conocimientos de informática y conocimientos avanzados sobre Android
Usuario estándar Android	2	Usuarios de móviles Android, que no poseen conocimientos avanzados de informática
Usuario Estándar Otras Plataformas.	1	Usuarios de SmartPhones con otros sistemas operativos, que no poseen conocimientos avanzados de informática

Ilustración 101 Tabla Perfiles Evaluador

7.1.2. Pruebas.

Las pruebas que se establecen para realizar el test de la aplicación se basan en los diferentes casos de usos que se han expuesto previamente.

Tarea	DESCRIPCIÓN
Registrarse	T1 El usuario debe completar de forma satisfactoria el proceso de registro por el medio que el elija.
Editar perfil.	T2 Se les solicitará a los usuarios que cambien su foto de perfil, su contraseña y su correo de contacto.
Jugar Solo	T3 Cada usuario debe ser capaz de jugar 2n la modalidad solo, al menos 3 partidas en un tiempo no mayor a 3 minutos.
Retarse entre ellos	T4 Se solicitará a cada usuario que rete a otro, para lo que previamente deben haber realizado una petición de amistad.
Decir el ranking propio y de amigos	T5 Cada usuario deberá ser capaz de saber el ranking de su listado de amigos.
Enviar una solicitud de compartir la app en cualquier medio	T6 Cada usuario deberá compartir un mensaje de la aplicación ya sea bien por WhatsApp, Facebook, Twitter, SMS o email. Vale tanto compartir como pedir ayuda en juego.

Ilustración 102 Tabla Pruebas a Realizar

La consecución de estas pruebas en unos tiempos prudenciales nos asegurarán que el diseño es el correcto, a su vez podremos ver donde se encuentran las mayores cáscaras de plátano que no hemos sabido ver y estableceremos un plan de mejoras, que aunque no se realice en esta primera fase serán de máxima prioridad en el apartado de mejoras futuras.

7.1.3. resultados.

El proceso de evaluación se realizará mediante la pregunta directa a el usuario de cómo ha percibido la complejidad de esa tarea englobándola bajo el siguiente baremo:

- 1- Muy difícil (MD)
- 2- Difícil (D)
- 3- Media (M)
- 4- Fácil (F)
- 5- Muy fácil (MF)

A su vez se evaluará el tiempo de realización, para las tareas con una evaluación de 4 ó 5 poseen desviación, hacia un mayor tiempo que la media de los demás usuarios, se le restará un punto de la nota.

Tras la realización de las pruebas, los resultados obtenidos han sido:

Tarea / Usuario		Experto		Estándar Android 1		Estándar Android 2		Estándar Otros 1	
		Dificultad - tiempo							
Registrarse	T1	F	183'	M	70'	M	96'	D	163'
Editar perfil.	T2	F	10'	F	7'	F	15'	F	9'
Jugar Solo	T3	MF	5'	MF	9'	MF	3'	MF	4'
Retarse entre ellos	T4	F	35'	M	31'	F	26'	D	42'
Decir el ranking propio y de amigos	T5	M	30'	MF	12'	MF	19'	F	13'
Enviar una solicitud de compartir la app en cualquier medio	T6	F	9'	F	11'	MF	17'	F	11'

Ilustración 103 Tabla Resultados Control de calidad

7.2. Modificaciones de mejora

Con los resultados en la mano hemos visto que la usabilidad de la aplicación es buena, no obstante hemos detectado ciertos fallos que deberemos proceder a subsanar.

- 1- El registro NO es claro, los usuarios se lía al hacer el registro pues todos han empezado a registrarse en la pantalla de LOGIN.
Para subsanar esta eventualidad se ha optado por clarificar el layout diferenciando claramente las dos partes.

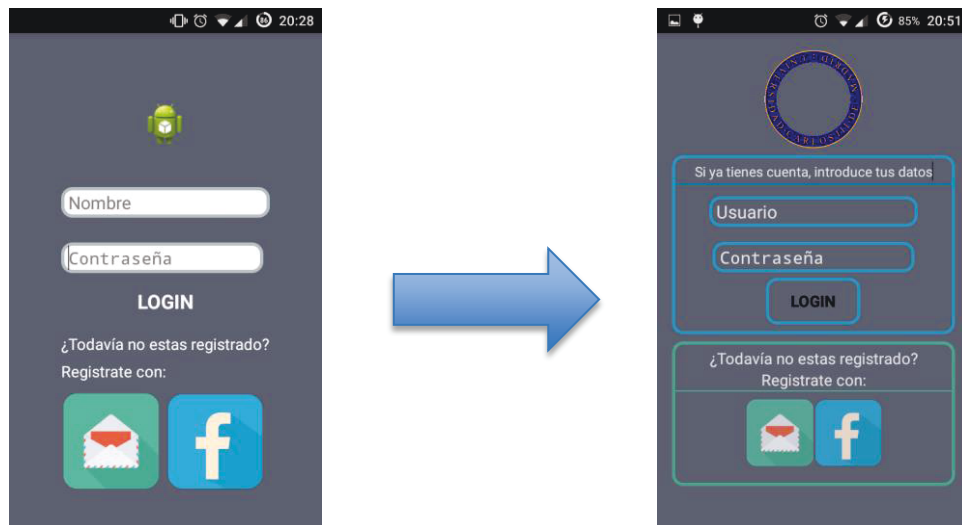


Ilustración 104 Modificación propuesta en control de calidad I

- 2- Al registrarse los cuadros de texto para escribir no se ven bien por los colores seleccionados en los Hint, se pasa a blanco, además se establece un nuevo código de colores más acorde con la aplicación.

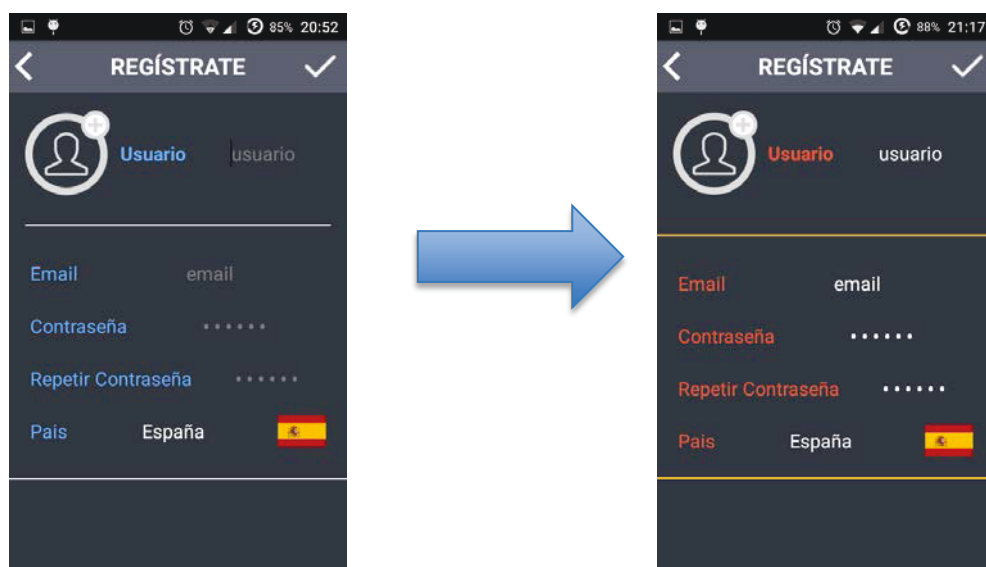


Ilustración 105 Modificación propuesta en control de calidad II

- 3- El teclado implementado necesita cierta practica hasta que el usuario coge pericia para jugar con soltura, sobre todo en móviles con pantallas reducidas. Esto debido a el rediseño del teclado lose estudiará en opciones futuras.
- 4- El botón de borrar en los primero juegos, no es fácilmente identificable, se procede a aumentar un poco su tamaño, y se cambia la ubicación para dejar clara la funcionalidad.
- 5- Las opciones del perfil para ver las estadísticas y los amigos no son del todo claras hasta que se ha llegado al menos una vez. Subsanan este fallo nos obligaría a hacer un remake completo de la app, y debido a la escasa muestra sobre la que hemos trabajado, tampoco se estima, por lo menos por el momento plantear un cambio de diseño hasta saber si es generalizado el problema y si este supone un problema al usuario.
- 6- Se podría pensar en buscar otras palabras a “estadísticas” y rankings pues la gente las confunde.
- 7- Al ir al perfil, no estaba suficientemente claro que la fotografía era un botón, se hace un completo rediseño de la pantalla inicial, detallando con el mismo tipo de borde todo lo que son los botones principales, A su vez se rediseña pensando en mejorar el aspecto visual imperante de la aplicación.

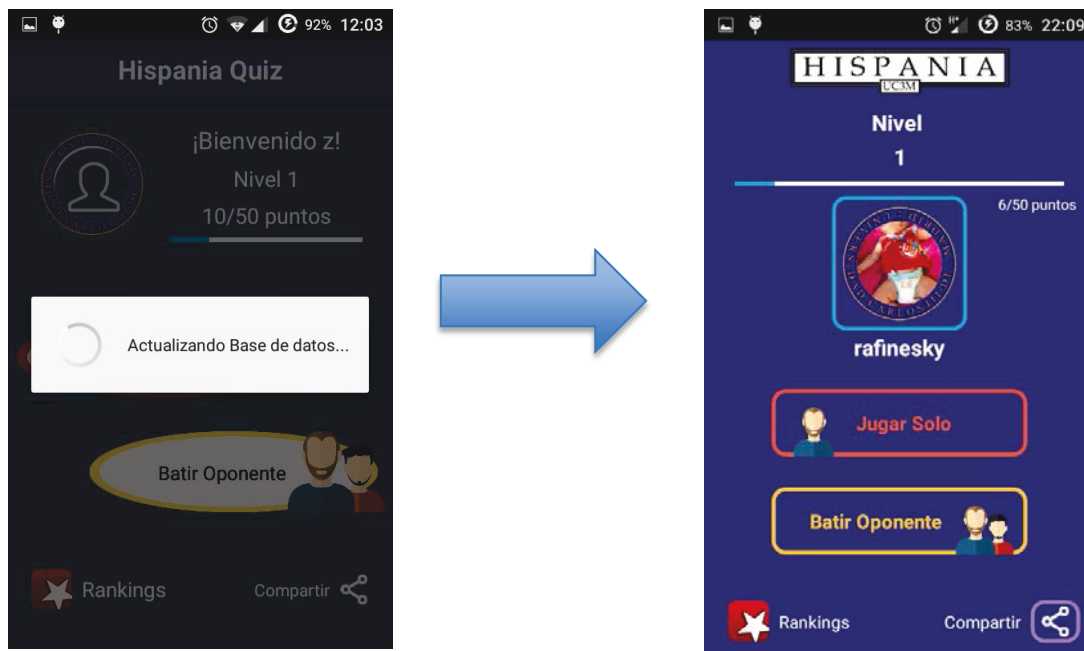


Ilustración 106 Modificación propuesta en control de calidad III

CAPÍTULO 8

PLANIFICACIÓN Y PRESUPUESTO

8. PLANIFICACIÓN Y PRESUPUESTO

En este capítulo se presentarán justificados los costes asociados a este proyecto, para lo cual nos guiaremos por la planificación llevada a cabo a lo largo del mismo, identificando las tareas a realizar el Rol desempeñado en cada una y asignando un costo a cada perfil sobre el que finalmente estipularemos la suma final.

8.1. Planificación

8.1.1. Identificación de Tareas.

Como se ha detallado previamente el proyecto se ha realizado mediante un desarrollo iterativo-incremental en 4 Fases, a las que habrá que sumar las operativas de liberación del código. En cada fase se siguen todas las fases típicas del desarrollos software:

Análisis: identificación de requisitos exigibles para el correcto desempeño de la aplicación.

Diseño: Establecer la solución que se estima correcta para la implementación acorde a los requisitos especificados.

Implementación: creación del código fuente en función del diseño propuesto.

Pruebas: Diseño y ejecución de un amplio abanico de pruebas basadas en las funcionalidades implementadas que aseguren una correcta validación del software realizado.

A su vez, al finalizar el proyecto se ha llevado a cabo un control de calidad y el correspondiente despliegue, con lo que también se detallarán estos en el presupuesto. Al tratarse de una aplicación móvil también propondremos un plan de mantenimiento y mejoras que se propongan sobre en siguiente capítulo.

Para cada tarea identificada se procederá a detallar el número de horas estimadas que necesita la tarea para verse completada

FASE	GRUPO	Id	TAREA	PERFIL		
				A	DG	P
Fase 0 Estudio Viabilidad y concepto.	Análisis	1	Extracción de concepto	6		
		2	Estudio viabilidad del proyecto y mercados potenciales	24		
Fase 1 sistemas básicos, control de usuario , conectividad backend y sistema de juego para 1 jugador.	Análisis	3	Mockups Papel / diseño de sistemas de juego	8		
		4	plataforma Android	8		
		5	Análisis sistemas backend	6		
	Diseño	6	Sistema de juego / negocio	12		
		7	Creación de interfaces		32	
	Implementación	8	Creación del código asociado a la lógica de negocio			92
		9	Programación App42			24
		10	Programación Parse.com			16
		11	Creación de Código Java asociado a 1 jugador			46
		12	rankings			20
		13	Vinculación del código a los backend			19
	Pruebas	14	Realización de pruebas asociadas a Fase 1.	3		
Fase 2 integración modo multijugador	Análisis	15	diseño de sistemas de juego	10		
	Diseño	16	Sistema de juego	8		
		17	Creación de vistas		14	
	Implementación	18	Creación del código asociado a 2 jugador			60
		19	Programación App42			12
		20	Programación Parse.com			0,5
		21	Módulo publicidad.			3
	Pruebas	22	Realización de pruebas asociadas a Fase 2.	3		
Fase 3 integración con redes sociales y Share de la app	Análisis	23	diseño de sistemas de compartición	4		
	Diseño	24	Actividades de compartir	2		
		25	Creación de vistas		10	
	Implementación	26	Facebook			32
		27	WhatsApp			1
		28	Twitter			4
		29	email			0,5
		30	SMS			0,5
	Pruebas	31	Realización de pruebas asociadas a Fase 3.	4		
Control de Calidad		32	Realización del control de calidad propuesto		4	
		33	Adecuación de la App en base a los resultados		10	8
Despliegue		34	Subida a Google Play , ficha, creación de cuenta...		4	

Ilustración 107 Tabla Tareas

A= Analista

DG=Diseñador gráfico

P= Programador

8.1.2. Diagrama de Gantt.

A continuación se muestran mediante Un diagrama de Gantt LA consecución y cronograma de las diferentes fases del proyecto. Se establece como una fecha de inicio hipotética el 1 de Septiembre de 2015, ayudando a valorar la envergadura y consunción de fases a lo largo del tiempo de vida del proyecto.

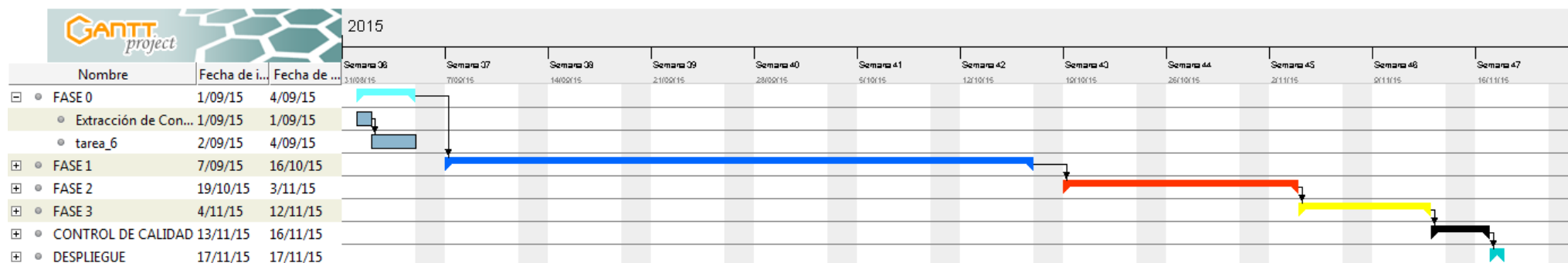


Ilustración 108 : Detalle Gantt Global

A continuación se mostrarán las secuencias desglosadas de cada tarea en cada una de las Fases de desarrollo.

FASE 0

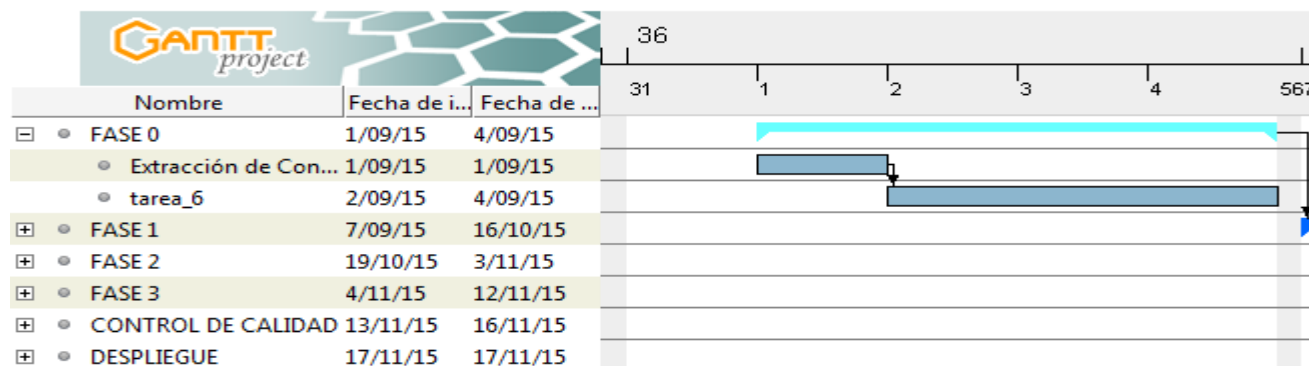


Ilustración 109 : Gantt Fase 0

FASE 1

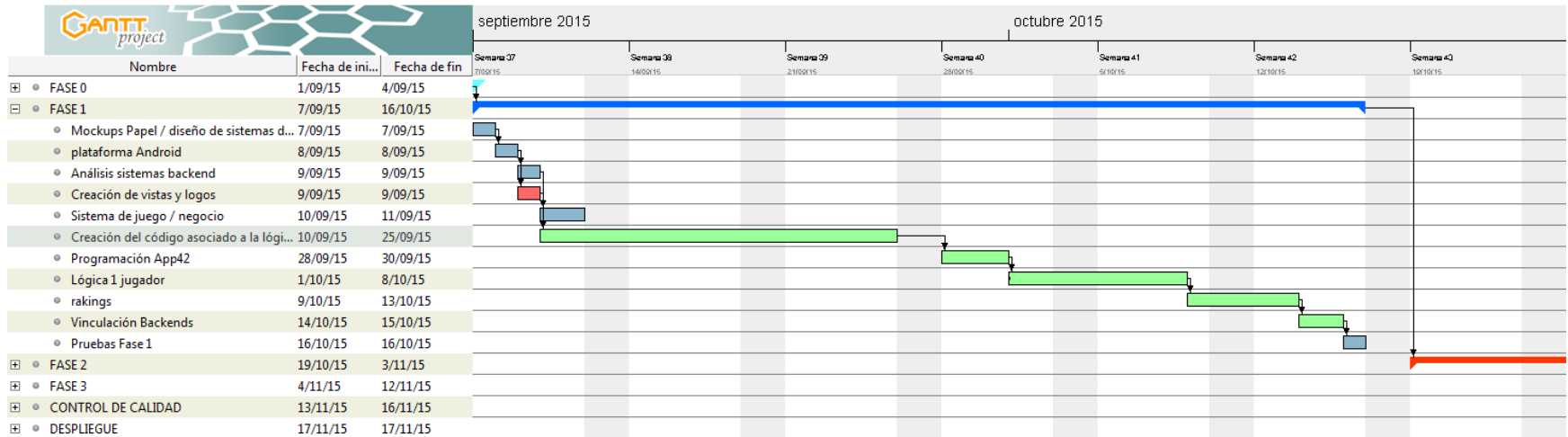


Ilustración 110 : Gantt Fase 1

FASE 2

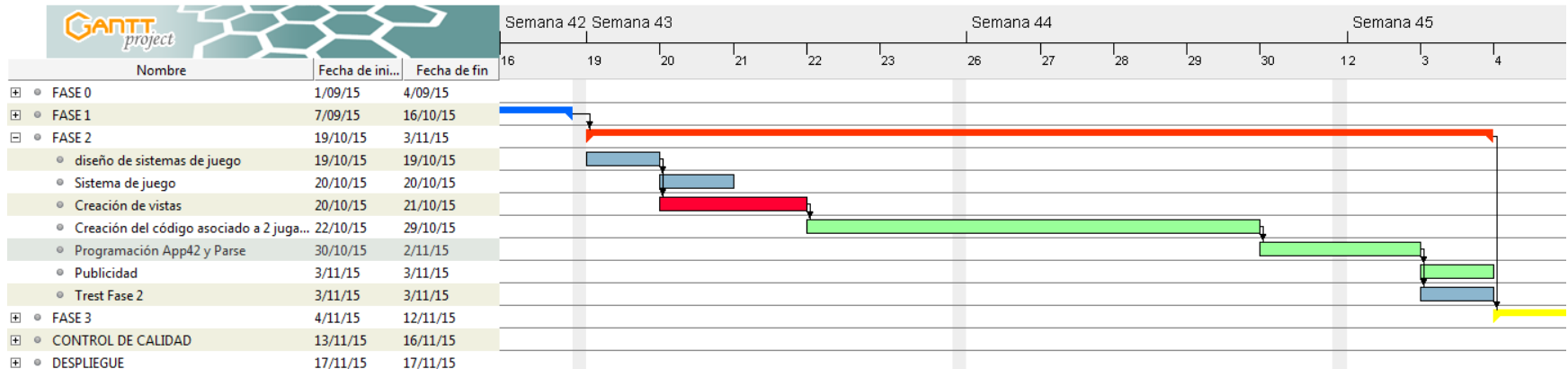


Ilustración 111 : Gantt Fase 2

FASE 3

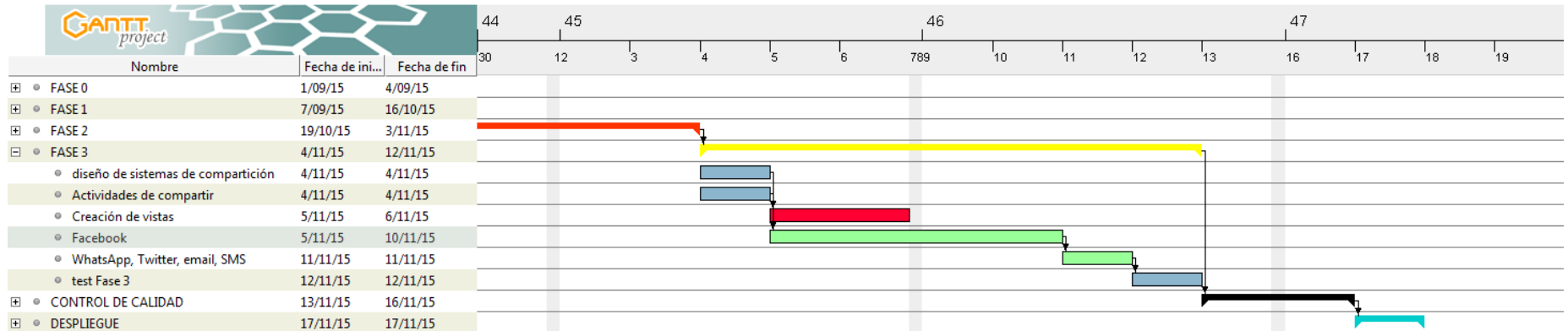


Ilustración 112 : Gantt Fase 3

CONTROL DE CALIDAD Y DESPLIEGUE:

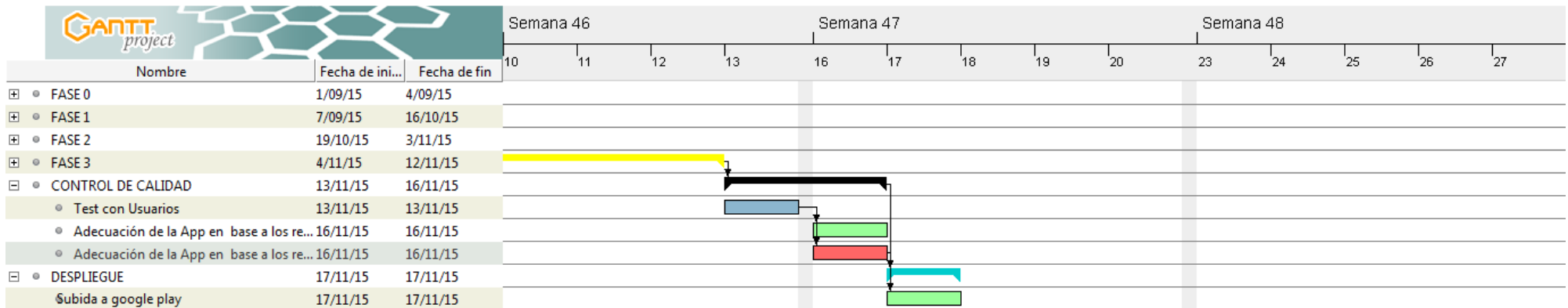


Ilustración 113 : Gantt Control de calidad y despliegue

8.1.3. Diagrama de Recursos

Finalmente y a modo de una adecuada previsión de los recursos en base a los objetivos se refleja el diagrama de Recursos, donde se ha establecido una unidad de los perfiles de programador, una unidad del perfil Analista y una unidad de Perfil Diseñador gráfico.

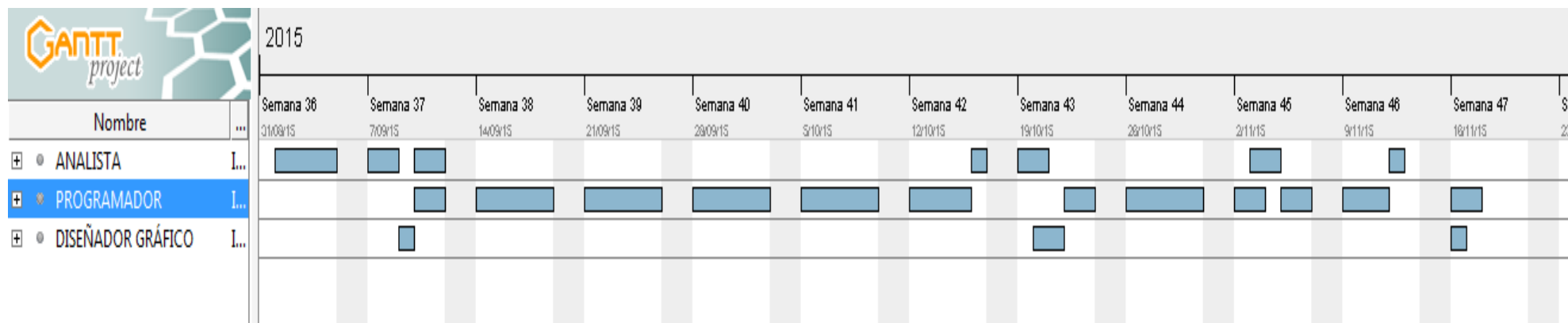


Ilustración 114 Diagrama de recursos

Se puede observar como para la consecución total del proyecto se estiman unas 12 semanas, desde el inicio de Fase de conceptualización y diseño, su puesta en producción, pruebas de calidad adecuación y entrega final. EL tiempo se podría ver acortado en torno a un 25% si se contase con un segundo recurso Programador, este podría realizar la implementación paralela de otra Fase, no obstante debido a la existencia de una fuerte serialidad en muchas tareas y por ejemplo la necesidad de la fase 3 de poseer las anteriores y la 1 y la 2 de haber completado la Fase 0, en el único momento que se establecería un paralelismo real sería al finalizar la Fase 0 que se podría implementar las fases q y 2 de manera paralela.

8.2. Presupuesto

En Base a las tareas previamente identificadas y a su coste en horas / recurso se realizar un presupuesto mediante la asignación de los costos de cada Perfil.

Perfil	Precio / Hora del recurso (SIN IVA)€
Analista	35
Programador	28
Diseñador Gráfico	30
Tester	20
Director de Proyecto	52

Ilustración 115 tabla Costes / Recurso

Debido a la envergadura del proyecto se ha optado por ser un coste pequeño por utilizar al analista como parte integrante del proceso de Testeo. El haber definido la aplicación y tener un profundo conocimiento sobre ella nos asegura que realizará todos los test de una manera eficiente, por lo que el aumento de precio por hora no incrementará apenas el presupuesto en comparación con la utilización de otro perfil menos preparado.

A su vez, al tratarse de un equipo pequeño No se estima necesaria la figura de Director de proyecto, habiéndose horas de Analista para poder llevar a cabo las tareas de coordinación que surjan en el proyecto.

Con estos datos de costo por recurso calculamos el precio de cada una de las Fases de implementación:

Tareas Fase 0	ANALISTA	GRAFICO	PROGRAMADOR
1	6		
2	24		
TOTAL FASE 0	30		

Ilustración 116 Recursos en Horas Fase 0

FASE 1	ANALISTA	GRAFICO	PROGRAMADOR
3	8		
4	8		
5	6		
6	12		
7		32	
8			92
9			24
10			16
11			46
12			20
13			19
14	3		
TOTAL FASE 1	37	32	217

Ilustración 117 Recursos en Horas Fase 1

Se puede observar como el arranque de la aplicación y el establecimiento de la lógica de negocio genera un mayor montante, esto es importante detallarlo, pues al tratarse de una aplicación móvil el componente visual es muy relevante a vistas del cliente y puede no comprender correctamente el porqué las diferencias de tiempos en cada fase y el numero de pantallas generadas.

FASE 2	ANALISTA	GRAFICO	PROGRAMADOR
15	10		
16	8		
17		14	
18			80
19			12
20			0,5
21			3
22	3		
TOTAL FASE 2	21	14	95,5

Ilustración 118 Recursos en Horas Fase 2

FASE 3	ANALISTA	GRAFICO	PROGRAMADOR
23	4		
24	2		
25		10	
26			32
27			1
28			4
29			0,5
30			0,5
31	4		
TOTAL FASE 3	8	10	

Ilustración 119 Recursos en Horas Fase 3

A continuación estimamos la fase de control de calidad, la cual debido a el gran esfuerzo realizado en el diseño, no se espera genera grandes o profundos cambios en el aplicativo.

CONTROL CALIDAD	ANALISTA	GRAFICO	PROGRAMADOR
32	4		
33		10	8
TOTAL Control Calidad	4	10	8

Ilustración 120 Recursos en Horas Control Calidad

DESPLIEGUE	ANALISTA	GRAFICO	PROGRAMADOR
25			4
TOTAL Despliegue			4

Ilustración 121 Recursos en Horas Despliegue

Una vez obtenidos los cálculos por Fase, se puede observar como en la primera, al tener que integrar el grosso de la lógica de negocio el montante es mayor, también lo es la necesidad de contar con recursos como el analista pues la criticidad de las tareas es más elevada y sentará las bases para poder agilizar los desarrollos en las siguientes fases.

En Base a estos resultados el Montante Final de la aplicación.

	ANALISTA	GRAFICO	PROGRAMADOR
HORAS TOTALES	110h	59h	409,5h
DIAS	13,75	7,375	51,1875
PRECIO RECURSO	35€/h	30€/h	28€/h
COSTE TOTAL RECURSO	3.850€	1.770€	11.466€

Ilustración 122 Tabla coste por recurso

	PRECIO SIN IVA	PRECIO CON IVA (21%)
COSTES TOTALES	17.086€	20.674,06€

Ilustración 123 Tabla Costes totales

ESTE PRESUPUESTO NO INCLUYE

- El pago de la creación de la Cuenta en google Play.
- Todos los pagos asociados a el exceso de uso sobre las capas gratuitas ofrecidos por los servicios de Back-end, en ningún caso nos hacemos responsables en los cambios de cuotas o políticas que en estos se pudieran suceder.

En caso de que se estime oportuno una vez entregado y validado el proyecto, el cliente contará con una **oferta de Soporte** por la cual, mediante el pago de una cuota mensual incluye:

1. Asegurar la compatibilidad con nuevas versiones de Android
2. La corrección de Bugs de vistas que pudieran aparecer en ciertos móviles que no hayan podido ser incluidos en los Test iniciales y adecuación a terminales futuros.
3. La actualización de los datos de la base de datos, siempre que el cliente las facilite en formato CSV con un máximo de 2 Mensuales.
4. Un descuento del 20% en las primeras 20 horas que puedan surgir en cuanto al desarrollo de mejoras o nuevas funcionalidades
5. Un resumen mensual sobre el uso de las capacidades de los Back-end y de la aplicación.

	SIN IVA	CON IVA (21%)
Programa Mantenimiento	400€ / mes	484€ / mes

Ilustración 124 Tabla Coste Mantenimiento

Validez y Aceptación de la oferta:

Las condiciones de la presente oferta son válidas hasta el 30/09/2015, la aceptación de dicha oferta se realizará por correo ordinario o correo electrónico a las personas implicadas en la realización y debe ser aceptada tanto por la parte involucrada en desarrollo como por la Universidad Carlos III de Madrid.

CAPÍTULO 9

Líneas de Desarrollo Futuras

9. LÍNEAS DE DESARROLLO FUTURAS

En esta sección, y una vez habiendo acabado nuestra aplicación me gustaría hacer una mención a una frase de Nelson Mandela:

“Después de escalar una montaña muy alta, descubrimos que hay muchas otras montañas por escalar”

Aplicando esta frase en conjunto con el afán de superación y mejora que posee intrínsecamente el ser humano, una vez se considera terminado un proyecto siempre queremos ver que nuevos pasos podemos dar para llevar nuestra base existente a nuevas metas. En este caso y debido a la naturaleza del proyecto las metas son mejoras o incorporación de nuevas funcionalidades a la aplicación existente o apertura de nuevos mercados que hagan crecer nuestra base de usuarios.

9.1. Apertura de nuevos mercados

La forma más fácil de abrir nuevos mercados que faciliten la ampliación de nuestra base de usuarios es portar nuestra aplicación actual a estos. Basándome en los estudios realizados en el capítulo 3 del presente documento se estima como primordial la generación de una App compatible con la actual en el mercado proporcionado por los Dispositivos Apple. Esta decisión se basa en la gran base de usuarios que ya tiene, así como el potente mercado de aplicaciones sobre el que podemos trabajar.

Como consideración se debería estudiar la integración completa como juego de Facebook, realizando una aplicación compatible para jugar desde la web de dicha red social.

Las implementaciones sobre otros sistemas operativos o plataformas se dejan descartadas debido a el escaso aporte que nos daría en comparación con el esfuerzo técnico y financiero que se debería hacer.

9.2. Mejoras sobre la infraestructura existente.

Una de las Grandes carencias que se han detectado , es la necesidad de nuestra aplicación a estar vinculada a el habla hispana. Sería recomendable hacer una traducción a otros idiomas e intentar ganar nichos de mercado sobre bases poblacionales interesadas en historia de España, no obstante nos veríamos obligados en previsión de un reducido número de usuarios a hacer una traducción muy literal de las preguntas, manteniendo así la compatibilidad de la base de datos independientemente del idioma elegido, es decir, permitir que dos usuarios con distinto idioma puedan competir conjuntamente.

Otro de los aspectos que se ha dejado fuera del proyecto ha sido la integración completa con Facebook, la salida de las nuevas Api nos han imposibilitado ciertas acciones como son enviar invitaciones a amigos que tengamos en Facebook para jugar a nuestro juego o poder retar a otros usuarios que estén validados con dicha red social de una manera directa sin tener que realizar un “juego de Facebook” , es decir un desarrollo totalmente enfocado a que la aplicación exista en Facebook y haga uso de su sistema propio de Juegos.

Siguiendo con mejoras sobre redes sociales, en esta versión no existe integración alguna con Google plus, su fácil programación lo hacen de un candidato ideal para ser una de las primeras mejoras a ser integradas.

Como evolución sobre el modelo de juego, se prevería incluir un modo síncrono de juego, en donde dos usuarios se batan por responder el mayor número de preguntas en el mismo tiempo establecido.

9.3. back-end

Los sistemas backend, tanto como mejora evolutiva como por necesidades económicas , nos podemos ver forzados a realizar un backend propio o cambiarlo de hospedaje en caso de que las condiciones contractuales de estos cambien o , en algún momento un cambio genere una mejora económica sobre nuestro modelo de negocio por poseer un elevado numero de usuarios concurrentes.

REFERENCIAS

10. Bibliografía y referencias

1. *Description by Google*. Available at: <http://www.google.es> (Accessed: 26 July 2015)
2. Varios:
 - 2.1 jonadep *Smartphones / Historia de la Informática*. Available at: <http://histinf.blogs.upv.es/2012/12/03/smartphones/> (Accessed: 26 July 2015)
 - 2.1. 2.2 Pastor, J. and Móvil, X. *Y el primer smartphone de la historia fue...* Available at: <http://www.xatakamovil.com/movil-y-sociedad/y-el-primer-smartphone-de-la-historia-fue> (Accessed: 26 July 2015)
 - 2.2. 2.3 A., C. M. 'Reloj', Available at: <http://smartphoneavancetecnologico.blogspot.com.es/p/historia-y-evolucion-del-smartphone.html> (Accessed: 26 July 2015)
3. Jayachandra, Y. *SMARTPHONE FRONTIERS*. McGraw-Hill Education
4. Microsoft, R. Available at: <http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/a/pdf/press%20release%201993.pdf> (Accessed: 26 July 2015)
5. Dan, M. (2014) *November 23, 1992: The First Smart Phone, the IBM Simon Personal Communicator, is Introduced*. Available at: <http://www.historyandheadlines.com/november-23-1992-first-smart-phone-ibm-simon-personal-communicator-introduced/> (Accessed: 26 July 2015)
6. Microsoft, R. *Detail*. Available at: <http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=40> (Accessed: 26 July 2015)
7. Zheng, P. and Ni, L. (2014) *Smart Phone and Next Generation Mobile Computing*. United States: Morgan Kaufmann Publishers
8. *History « Stockholm Smartphone*. Available at: <http://www.stockholmsmartphone.org/history/> (Accessed: 26 July 2015)
9. Available at: <https://www.wayerless.com/2011/07/ericsson-r380-el-primer-smartphone-exitoso/> (Accessed: 26 July 2015)
10. Hand, C. (2015) *How the Internet Changed History*. Edited by ABDO.
11. (2015) *The history of BlackBerry: in pictures*. Available at: <http://www.telegraph.co.uk/technology/blackberry/11347347/The-history-of-BlackBerry-in-pictures.html> (Accessed: 26 July 2015)
12. Fingas, J. *RIM: A brief history from Budgie to BlackBerry 10*. Available at: <http://www.engadget.com/2013/01/28/rim-a-brief-history-from-budgie-to-blackberry-10/> (Accessed: 26 July 2015)
13. *Historia del smartphone que lo cambió todo, el Iphone*. | Grupo Bonatel S.L.U. Distribuidor Autorizado Vodafone. Available at: <http://www.grupobonatel.com/iphone-cumple-7-anos/> (Accessed: 26 July 2015)
14. Naber, T. (2015) *How the Computer Changed History*. Edited by ABDO.

15. tomwarren (2014) *iPhone: A visual history*. Available at: <http://www.theverge.com/2014/9/9/6125849/iphone-history-pictures> (Accessed: 26 July 2015)
16. Zak, R. *The most game-changing phones in Android history*. Available at: <https://www.androidpit.com/most-important-android-phones-in-history> (Accessed: 26 July 2015)
17. Perez, I. L. (2015) *La historia de Android: de Cupcake a M*. Available at: <http://www.omicrono.com/2015/06/historia-de-android/> (Accessed: 26 July 2015)
18. Girones, J. T. and Vogt, C. (2015) *Hands on Android*. United States: Marcombo S.A.
19. (2011) *La historia de cómo Android hizo que Motorola vuelva a las 'grandes ligas'*. Available at: <http://www.redusers.com/noticias/como-hizo-android-que-motorola-vuelva-a-las-%E2%80%9Cgrandes-ligas%E2%80%9D/> (Accessed: 26 July 2015)
20. Congress, 56 nereanietoMobile World, Galaxy, S., S. G. and 5, S. G. S. (2014) 'La evolución de Samsung hasta llegar al Galaxy S5',
21. *A Brief History of the Samsung Galaxy S Series*. Available at: <http://www.mobilemadhouse.co.uk/history-of-samsung-galaxy-s-series> (Accessed: 26 July 2015)
22. Samsung, O. (2007) *Welcome to a Superior Life - I9000 Galaxy S - Pantalla Táctil - Teléfonos Móviles /*. Available at: http://www.samsung.com/es/test_es.html (Accessed: 26 July 2015)
23. Samsung, A. (2011) *Samsung Galaxy S2 a fondo / SAMSUNG Argentina*. Available at: <http://www.samsung.com/ar/news/local/23914> (Accessed: 26 July 2015)
24. Samsung *Samsung GALAXY S III*. Available at: <http://www.samsung.com/latam/galaxys3/specifications.html> (Accessed: 26 July 2015)
25. Samsung *Smartphone Galaxy S4 I9515 - INFORMACIÓN GENERAL*. Available at: <http://www.samsung.com/es/consumer/mobile-devices/smartphones/galaxy-s/GT-I9515ZSAPHE> (Accessed: 26 July 2015)
26. Samsung *My life powered by GalaxyS5*. Available at: <http://www.samsung.com/es/offer/galaxys5/> (Accessed: 26 July 2015)
27. *Samsung Galaxy S6 - Características*. Available at: <http://www.samsung.com/es/consumer/mobile-devices/smartphones/galaxy-s/SM-G920FZDAPHE> (Accessed: 26 July 2015)
28. *Arquitectura ARM (procesadores)*. Available at: <http://www.importancia.org/arquitectura-arm.php> (Accessed: 26 July 2015)
29. Espeso, P. *ARM, la 'navaja suiza' de los procesadores*. Available at: <http://www.xataka.com/componentes/arm-la-navaja-suiza-de-los-procesadores-1> (Accessed: 26 July 2015)
30. Available at: http://docsetools.com/articulos-noticias-consejos/article_126634.html (Accessed: 26 July 2015)
31. Mark (2012) *The Crazy Story Of BitBoys - Mobile: Intel Will Overtake Qualcomm In Three Years*. Available at: <http://www.tomshardware.com/reviews/medfield-krait-smartphone-mobile-soc,3117-7.html> (Accessed: 26 July 2015)
32. Espeso, P. (2015) *¿Hasta dónde va a llevar ARM al smartphone en 2015 con el nuevo Cortex-A72?*. Available at: <http://www.xataka.com/moviles/arm-cortex-a72-mas-lena-para-los-futuros-smartphones-y-tablets-mas-premium> (Accessed: 26 July 2015)

33. Jayachandra, Y. *SMARTPHONE FRONTIERS*. McGraw-Hill Education
34. Perez, I. L. (2015) *La historia de Android: de Cupcake a M*. Available at: <http://www.omicrono.com/2015/06/historia-de-android/> (Accessed: 26 July 2015)
35. jonadep *Smartphones / Historia de la Informática*. Available at: <http://histinf.blogs.upv.es/2012/12/03/smartphones/#mali> (Accessed: 26 July 2015)
36. Lequerica, J. R. (2011) *Manual imprescindible de desarrollo de aplicaciones para Android*. Madrid: Anaya Multimedia
37. Pérez, J. C. (2007) *Sistemas operativos: una visión aplicada*. 2nd edn. Madrid: McGraw-Hill / Interamericana de España
38. 2.2. *Arquitectura Android - Software de Comunicaciones*. Available at: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android> (Accessed: 26 July 2015)
39. Wiki, S. (2013) *Mach: Darwin (Operating System), Gnu-Darwin, Gnu Mach, IOS, Mach-O, Machten, Macmach, Mtxinu, NeXTSTEP, OS X, Rhapsody (Oper.* United States: University-Press.Org
40. Michán, M. and Applesfera (2013) *La evolución de iOS desde 2007 hasta la actualidad [Especial Historia WWDC]*. Available at: <http://www.applesfera.com/ios/la-evolucion-de-ios-desde-2007-hasta-la-actualidad-especial-historia-wwdc> (Accessed: 26 July 2015)
41. 2.1. *Características - Software de Comunicaciones*. Available at: <https://sites.google.com/site/swcuc3m/home/iphone/2-generalidades-1/2-1-caracteristicas> (Accessed: 26 July 2015)
42. Available at: http://docsetools.com/articulos-noticias-consejos/article_127071.html (Accessed: 26 July 2015)
43. Pérez, J. C. (2007) *Sistemas operativos: una visión aplicada*. 2nd edn. Madrid: McGraw-Hill / Interamericana de España
44. Porras, E. *Ingeniería de sistemas*. Available at: <http://eve-ingsistemas-u.blogspot.com.es/2012/04/sistemas-operativos-moviles-ios.html> (Accessed: 26 July 2015)
45. Available at: <http://www.chw.net/2012/06/que-significa-que-windows-phone-8-comparta-su-nucleo-con-windows-8/> (Accessed: 26 July 2015)
46. Arredondo, J. L. (2015) *Windows Phone pierde más de la mitad de su cuota de mercado en España*. Available at: http://cincodias.com/cincodias/2015/03/05/smartphones/1425567370_450919.html (Accessed: 26 July 2015)
47. Ion, F. *How BlackBerry ported over QNX for its new OS*. Available at: <http://arstechnica.com/gadgets/2013/02/from-the-car-to-your-phone-how-blackberry-porting-over-qnx-for-its-new-os/> (Accessed: 26 July 2015)
48. IDC: *Smartphone OS Market Share*. Available at: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> (Accessed: 26 July 2015)
49. (2015) *Arquitectura del Sistema Operativo Firefox OS*. Available at: https://developer.mozilla.org/es/Firefox_OS/Architecture (Accessed: 26 July 2015)

50. Firefox OS. Available at: <https://www.mozilla.org/es-ES/firefox/os/2.0/> (Accessed: 26 July 2015)
51. Castro, A. and Tizen (2014) *¿Qué es Tizen y qué podemos esperar de este nuevo SO?*. Available at: <http://computerhoy.com/noticias/software/que-es-tizen-que-podemos-esperar-este-nuevo-9265> (Accessed: 26 July 2015)
52. Oyandel, J. P. *Tizen, Ligero y veloz*. Available at: <https://www.wayerless.com/2015/01/sistema-operativo-tizen-demuestra-ser-muy-ligero-y-veloz-en-de-equipos-gama-baja/> (Accessed: 26 July 2015)
53. Lansdell (2015) *Tizen for Dummies 2E*.
54. AMAZON, I. *Documentation - Amazon Apps & Games Developer Portal*. Available at: <https://developer.amazon.com/public/resources/learning-center/documentation> (Accessed: 26 July 2015)
55. Speculations, G. *Amazon Appstore Is More Lucrative For Developers Than Google Play Store*. Available at: <http://www.forbes.com/sites/greatspeculations/2012/04/03/amazon-appstore-is-more-lucrative-for-developers-than-google-play-store/> (Accessed: 26 July 2015)
56. Callaham, J. (2015) *Windows Store hits 200,000 app mark, with 385,000 apps in Windows Phone Store*. Available at: <http://www.windowscentral.com/windows-store-hits-200000-app-mark-385000-apps-windows-phone-store> (Accessed: 26 July 2015)
57. calendamaia and Dev, G. *Eclipse IDE*. Available at: <http://www.genbetadev.com/herramientas/eclipse-ide> (Accessed: 26 July 2015)
58. *Arquitectura de Android - Diploma de Especialización en desarrollo de aplicaciones para Android*. Available at: <http://www.androidcurso.com/index.php/tutoriales-android/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android> (Accessed: 26 July 2015)
59. Albeniz, I. (2011) *Archivos DEX (Dalvik Executable): Introducción*. Available at: <http://www.ikeralbeniz.net/2011/01/26/archivos-dex-dalvik-executable-introduccion/> (Accessed: 26 July 2015)
60. Vugec, F. *ART vs Dalvik - introducing the new Android runtime in KitKat*. Available at: <https://www.infinum.co/the-capsized-eight/articles/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat> (Accessed: 26 July 2015)
61. Tristan, J. (2001) *Android - ADA Framework (parte I)*. Available at: <http://www.javahispano.org/android/2013/1/9/tutorial-ada-framework-parte-i.html> (Accessed: 26 July 2015)
62. Ballesteros, T. (2014) *DROIDCON SPAIN 2015 CONFERENCE (como asistente)*. Available at: <https://github.com/mobandme/ADA-Framework> (Accessed: 26 July 2015)
63. PARSE, inc. Available at: <https://parse.com/docs/android/guide> (Accessed: 26 July 2015)
64. Google, inc. *Lista de funciones de Google Analytics – Google Analytics*. Available at: http://www.google.es/intl/es_ALL/analytics/features/ (Accessed: 26 July 2015)
65. Gordillo, A. L. (2013) *Lleva al éxito tu App Android con Google Analytics (Parte 2)*. Available at: <http://www.limecreativelabs.com/lleva-exito-app-android-google-analytics-parte-2/> (Accessed: 26 July 2015)
66. Villa, M. (2015) *Google Analytics*. Available at: <http://proyectosbds.com/blog/tutorial-google-analytics-20-de-20/> (Accessed: 26 July 2015)

APENDICES

11. APENDICE I – LICENCIA CREATIVE COMMONS

Creative Commons

Reconocimiento-CompartirIgual 2.5 España

CREATIVE COMMONS CORPORATION NO ES UN DESPACHO DE ABOGADOS Y NO PROPORCIONA SERVICIOS JURÍDICOS. LA DISTRIBUCIÓN DE ESTA LICENCIA NO CREA UNA RELACIÓN ABOGADO-CLIENTE. CREATIVE COMMONS PROPORCIONA ESTA INFORMACIÓN TAL CUAL (ON AN "AS-IS BASIS"). CREATIVE COMMONS NO OFRECE GARANTÍA ALGUNA RESPECTO DE LA INFORMACIÓN PROPORCIONADA, NI ASUME RESPONSABILIDAD ALGUNA POR DAÑOS PRODUCIDOS A CONSECUENCIA DE SU USO.

Licencia

LA OBRA (SEGÚN SE DEFINE MÁS ADELANTE) SE PROPORCIONA BAJO LOS TÉRMINOS DE ESTA LICENCIA PÚBLICA DE CREATIVE COMMONS ("CCPL" O "LICENCIA"). LA OBRA SE ENCUENTRA PROTEGIDA POR LA LEY ESPAÑOLA DE PROPIEDAD INTELECTUAL Y/O CUALESQUIERA OTRAS NORMAS RESULTEN DE APLICACIÓN. QUEDA PROHIBIDO CUALQUIER USO DE LA OBRA DIFERENTE A LO AUTORIZADO BAJO ESTA LICENCIA O LO DISPUESTO EN LAS LEYES DE PROPIEDAD INTELECTUAL.

MEDIANTE EL EJERCICIO DE CUALQUIER DERECHO SOBRE LA OBRA, USTED ACEPTA Y CONSIENTE LAS LIMITACIONES Y OBLIGACIONES DE ESTA LICENCIA. EL LICENCIADOR LE CEDE LOS DERECHOS CONTENIDOS EN ESTA LICENCIA, SIEMPRE QUE USTED ACEPTÉ LOS PRESENTES TÉRMINOS Y CONDICIONES.

1. Definiciones

- a) La "obra" es la creación literaria, artística o científica ofrecida bajo los términos de esta licencia.

- b) El “**autor**” es la persona o la entidad que creó la obra.
- c) Se considerará “**obra conjunta**” aquella susceptible de ser incluida en alguna de las siguientes categorías:
- 1) “**Obra en colaboración**”, entendiéndose por tal aquella que sea resultado unitario de la colaboración de varios autores.
 - 2) “**Obra colectiva**”, entendiéndose por tal la creada por la iniciativa y bajo la coordinación de una persona natural o jurídica que la edite y divulgue bajo su nombre y que esté constituida por la reunión de aportaciones de diferentes autores cuya contribución personal se funde en una creación única y autónoma, para la cual haya sido concebida sin que sea posible atribuir separadamente a cualquiera de ellos un derecho sobre el conjunto de la obra realizada.
 - 3) “**Obra compuesta e independiente**”, entendiéndose por tal la obra nueva que incorpore una obra preexistente sin la colaboración del autor de esta última.
- d) Se considerarán “**obras derivadas**” aquellas que se encuentren basadas en una obra o en una obra y otras preexistentes, tales como: las traducciones y adaptaciones; las revisiones, actualizaciones y anotaciones; los compendios, resúmenes y extractos; los arreglos musicales y, en general, cualesquiera transformaciones de una obra literaria, artística o científica, salvo que la obra resultante tenga el carácter de obra conjunta en cuyo caso no será considerada como una obra derivada a los efectos de esta licencia. Para evitar la duda, si la obra consiste en una composición musical o grabación de sonidos, la sincronización temporal de la obra con una imagen en movimiento (“synching”) será considerada como una obra derivada a los efectos de esta licencia.
- e) Tendrán la consideración de “**obras audiovisuales**” las creaciones expresadas mediante una serie de imágenes asociadas, con o sin sonorización incorporada, así como las composiciones musicales, que estén destinadas esencialmente a ser mostradas a través de aparatos de proyección o por cualquier otro medio de comunicación pública de la imagen y del sonido, con independencia de la naturaleza de los soportes materiales de dichas obras.
- f) El “**licenciador**” es la persona o la entidad que ofrece la obra bajo los términos de esta licencia y le cede los derechos de explotación de la misma conforme a lo dispuesto

en ella.

- g) “**Usted**” es la persona o la entidad que ejercita los derechos cedidos mediante esta licencia y que no ha violado previamente los términos de la misma con respecto a la obra, o que ha recibido el permiso expreso del licenciador de ejercitar los derechos cedidos mediante esta licencia a pesar de una violación anterior.
- h) La “**transformación**” de una obra comprende su traducción, adaptación y cualquier otra modificación en su forma de la que se derive una obra diferente. Cuando se trate de una base de datos según se define más adelante, se considerará también transformación la reordenación de la misma. La creación resultante de la transformación de una obra tendrá la consideración de obra derivada.
- i) Se entiende por “**reproducción**” la fijación de la obra en un medio que permita su comunicación y la obtención de copias de toda o parte de ella.
- j) Se entiende por “**distribución**” la puesta a disposición del público del original o copias de la obra mediante su venta, alquiler, préstamo o de cualquier otra forma.
- k) Se entenderá por “**comunicación pública**” todo acto por el cual una pluralidad de personas pueda tener acceso a la obra sin previa distribución de ejemplares a cada una de ellas. No se considerará pública la comunicación cuando se celebre dentro de un ámbito estrictamente doméstico que no esté integrado o conectado a una red de difusión de cualquier tipo. A efectos de esta licencia se considerará comunicación pública la puesta a disposición del público de la obra por procedimientos alámbricos o inalámbricos, incluida la puesta a disposición del público de la obra de tal forma que cualquier persona pueda acceder a ella desde el lugar y en el momento que elija.
- l) La “**explotación**” de la obra comprende su reproducción, distribución, comunicación pública y transformación.
- m) Tendrán la consideración de “**bases de datos**” las colecciones de obras ajenas, de datos o de otros elementos independientes como las antologías y las bases de datos propiamente dichas que por la selección o disposición de sus contenidos constituyan creaciones intelectuales, sin perjuicio, en su caso, de los derechos que pudieran subsistir sobre dichos contenidos.
- n) Los “**elementos de la licencia**” son las características principales de la licencia según la selección efectuada por el licenciador e indicadas en el título de esta licencia:

Reconocimiento de autoría (Reconocimiento), Compartir de manera igual (CompartirIgual).

2. **Límites y uso legítimo de los derechos.** Nada en esta licencia pretende reducir o restringir cualesquiera límites legales de los derechos exclusivos del titular de los derechos de propiedad intelectual de acuerdo con la Ley de Propiedad Intelectual o cualesquiera otras leyes aplicables, ya sean derivados de usos legítimos, tales como el derecho de copia privada o el derecho a cita, u otras limitaciones como la derivada de la primera venta de ejemplares.
3. **Concesión de licencia.** Conforme a los términos y a las condiciones de esta licencia, el licenciador concede (durante toda la vigencia de los derechos de propiedad intelectual) una licencia de ámbito mundial, sin derecho de remuneración, no exclusiva e indefinida que incluye la cesión de los siguientes derechos:
 - a) Derecho de reproducción, distribución y comunicación pública sobre la obra.
 - b) Derecho a incorporarla en una o más obras conjuntas o bases de datos y para su reproducción en tanto que incorporada a dichas obras conjuntas o bases de datos.
 - c) Derecho para efectuar cualquier transformación sobre la obra y crear y reproducir obras derivadas.
 - d) Derecho de distribución y comunicación pública de copias o grabaciones de la obra, como incorporada a obras conjuntas o bases de datos.
 - e) Derecho de distribución y comunicación pública de copias o grabaciones de la obra, por medio de una obra derivada.
 - f) Para evitar la duda, sin perjuicio de la preceptiva autorización del licenciador, y especialmente cuando la obra se trate de una obra audiovisual, el licenciador renuncia al derecho exclusivo a percibir, tanto individualmente como mediante una entidad de gestión de derechos, o varias, (por ejemplo: SGAE, Dama, VEGAP), los derechos de explotación de la obra, así como los derivados de obras derivadas, conjuntas o bases de datos, si dicha explotación pretende principalmente o se encuentra dirigida hacia la obtención de un beneficio mercantil o la remuneración monetaria privada.

Los anteriores derechos se pueden ejercitar en todos los medios y formatos, tangibles o intangibles, conocidos o por conocer. Los derechos mencionados incluyen el derecho a

efectuar las modificaciones que sean precisas técnicamente para el ejercicio de los derechos en otros medios y formatos. Todos los derechos no cedidos expresamente por el licenciador quedan reservados.

4. Restricciones. La cesión de derechos que supone esta licencia se encuentra sujeta y limitada a las restricciones siguientes:

- a) Usted puede reproducir, distribuir o comunicar públicamente la obra solamente bajo los términos de esta licencia y debe incluir una copia de la misma, o su Identificador Uniforme de Recurso (URI), con cada copia o grabación de la obra que usted reproduzca, distribuya o comunique públicamente. Usted no puede ofrecer o imponer ningún término sobre la obra que altere o restrinja los términos de esta licencia o el ejercicio de sus derechos por parte de los cesionarios de la misma. Usted no puede sublicenciar la obra. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la ausencia de garantías. Usted no puede reproducir, distribuir o comunicar públicamente la obra con medidas tecnológicas que controlen el acceso o uso de la obra de una manera contraria a los términos de esta licencia. Lo anterior se aplica a una obra en tanto que incorporada a una obra conjunta o base de datos, pero no implica que éstas, al margen de la obra objeto de esta licencia, tengan que estar sujetas a los términos de la misma. Si usted crea una obra conjunta o base de datos, previa comunicación del licenciador, usted deberá quitar de la obra conjunta o base de datos cualquier crédito requerido en el apartado 4c, según lo que se le requiera y en la medida de lo posible. Si usted crea una obra derivada, previa comunicación del licenciador, usted deberá quitar de la obra derivada cualquier crédito requerido en el apartado 4c, según lo que se le requiera y en la medida de lo posible.
- b) Usted puede reproducir, distribuir o comunicar públicamente una obra derivada solamente bajo los términos de esta licencia, o de una versión posterior de esta licencia con sus mismos elementos principales, o de una licencia iCommons de Creative Commons que contenga los mismos elementos principales que esta licencia (ejemplo: Reconocimiento-CompartirIgual 2.5 Japón). Usted debe incluir una copia de la esta licencia o de la mencionada anteriormente, o bien su Identificador Uniforme de Recurso (URI), con cada copia o grabación de la obra que usted reproduzca, distribuya o comunique públicamente. Usted no puede ofrecer o imponer ningún término

respecto de las obras derivadas o sus transformaciones que alteren o restrinjan los términos de esta licencia o el ejercicio de sus derechos por parte de los cesionarios de la misma. Usted debe mantener intactos todos los avisos que se refieran a esta licencia y a la ausencia de garantías. Usted no puede reproducir, distribuir o comunicar públicamente la obra derivada con medidas tecnológicas que controlen el acceso o uso de la obra de una manera contraria a los términos de esta licencia. Lo anterior se aplica a una obra derivada en tanto que incorporada a una obra conjunta o base de datos, pero no implica que éstas, al margen de la obra objeto de esta licencia, tengan que estar sujetas a los términos de esta licencia.

- c) Si usted reproduce, distribuye o comunica públicamente la obra o cualquier obra derivada, conjunta o base de datos que la incorpore, usted debe mantener intactos todos los avisos sobre la propiedad intelectual de la obra y reconocer al autor original, de manera razonable conforme al medio o a los medios que usted esté utilizando, indicando el nombre (o el seudónimo, en su caso) del autor original si es facilitado, y/o reconocer a aquellas partes (por ejemplo: institución, publicación, revista) que el autor original y/o el licenciador designen para ser reconocidos en el aviso legal, las condiciones de uso, o de cualquier otra manera razonable; el título de la obra si es facilitado; de manera razonable, el Identificador Uniforme de Recurso (URI), si existe, que el licenciador especifica para ser vinculado a la obra, a menos que tal URI no se refiera al aviso sobre propiedad intelectual o a la información sobre la licencia de la obra; y en el caso de una obra derivada, un aviso que identifique el uso de la obra en la obra derivada (e.g., “traducción castellana de la obra de Autor Original,” o “guión basado en obra original de Autor Original”). Tal aviso se puede desarrollar de cualquier manera razonable; con tal de que, sin embargo, en el caso de una obra derivada, conjunta o base de datos, aparezca como mínimo este aviso allá donde aparezcan los avisos correspondientes a otros autores y de forma comparable a los mismos.
- d) En el caso de la inclusión de la obra en alguna base de datos o recopilación, el propietario o el gestor de la base de datos deberá renunciar a cualquier derecho relacionado con esta inclusión y concerniente a los usos de la obra una vez extraída de las bases de datos, ya sea de manera individual o conjuntamente con otros materiales.

5. Exoneración de responsabilidad

A MENOS QUE SE ACUERDE MUTUAMENTE ENTRE LAS PARTES, EL LICENCIADOR OFRECE LA OBRA TAL CUAL (ON AN “AS-IS” BASIS) Y NO CONFIERE NINGUNA GARANTÍA DE CUALQUIER TIPO RESPECTO DE LA OBRA O DE LA PRESENCIA O AUSENCIA DE ERRORES QUE PUEDAN O NO SER DESCUBIERTOS. ALGUNAS JURISDICCIONES NO PERMITEN LA EXCLUSIÓN DE TALES GARANTÍAS, POR LO QUE TAL EXCLUSIÓN PUEDE NO SER DE APLICACIÓN A USTED.

6. Limitación de responsabilidad.

SALVO QUE LO DISPONGA EXPRESA E IMPERATIVAMENTE LA LEY APLICABLE, EN NINGÚN CASO EL LICENCIADOR SERÁ RESPONSABLE ANTE USTED POR CUALQUIER TEORÍA LEGAL DE CUALESQUIERA DAÑOS RESULTANTES, GENERALES O ESPECIALES (INCLUIDO EL DAÑO EMERGENTE Y EL LUCRO CESANTE), FORTUITOS O CAUSALES, DIRECTOS O INDIRECTOS, PRODUCIDOS EN CONEXIÓN CON ESTA LICENCIA O EL USO DE LA OBRA, INCLUSO SI EL LICENCIADOR HUBIERA SIDO INFORMADO DE LA POSIBILIDAD DE TALES DAÑOS.

7. Finalización de la licencia

- a) Esta licencia y la cesión de los derechos que contiene terminarán automáticamente en caso de cualquier incumplimiento de los términos de la misma. Las personas o entidades que hayan recibido obras derivadas, conjuntas o bases de datos de usted bajo esta licencia, sin embargo, no verán sus licencias finalizadas, siempre que tales personas o entidades se mantengan en el cumplimiento íntegro de esta licencia. Las secciones 1, 2, 5, 6, 7 y 8 permanecerán vigentes pese a cualquier finalización de esta licencia.
- b) Conforme a las condiciones y términos anteriores, la cesión de derechos de esta licencia es perpetua (durante toda la vigencia de los derechos de propiedad intelectual aplicables a la obra). A pesar de lo anterior, el licenciador se reserva el derecho a divulgar o publicar la obra en condiciones distintas a las presentes, o de retirar la obra en cualquier momento. No obstante, ello no supondrá dar por concluida esta licencia (o cualquier otra licencia que haya sido concedida, o sea necesario ser concedida, bajo los términos de esta licencia), que continuará vigente y con efectos completos a

no ser que haya finalizado conforme a lo establecido anteriormente.

8. Miscelánea

- a) Cada vez que usted explote de alguna forma la obra, o una obra conjunta o una base datos que la incorpore, el licenciador original ofrece a los terceros y sucesivos licenciarios la cesión de derechos sobre la obra en las mismas condiciones y términos que la licencia concedida a usted.
- b) Cada vez que usted explote de alguna forma una obra derivada, el licenciador original ofrece a los terceros y sucesivos licenciarios la cesión de derechos sobre la obra original en las mismas condiciones y términos que la licencia concedida a usted.
- c) Si alguna disposición de esta licencia resulta inválida o inaplicable según la Ley vigente, ello no afectará la validez o aplicabilidad del resto de los términos de esta licencia y, sin ninguna acción adicional por cualquiera las partes de este acuerdo, tal disposición se entenderá reformada en lo estrictamente necesario para hacer que tal disposición sea válida y ejecutiva.
- d) No se entenderá que existe renuncia respecto de algún término o disposición de esta licencia, ni que se consiente violación alguna de la misma, a menos que tal renuncia o consentimiento figure por escrito y lleve la firma de la parte que renuncie o consienta.
- e) Esta licencia constituye el acuerdo pleno entre las partes con respecto a la obra objeto de la licencia. No caben interpretaciones, acuerdos o términos con respecto a la obra que no se encuentren expresamente especificados en la presente licencia. El licenciador no estará obligado por ninguna disposición complementaria que pueda aparecer en cualquier comunicación de usted. Esta licencia no se puede modificar sin el mutuo acuerdo por escrito entre el licenciador y usted.

Creative Commons no es parte de esta licencia, y no ofrece ninguna garantía en relación con la obra. Creative Commons no será responsable frente a usted o a cualquier parte, por cualquier teoría legal de cualesquiera daños resultantes, incluyendo, pero no limitado, daños generales o especiales (incluido el daño emergente y el lucro cesante), fortuitos o causales, en conexión con esta licencia. A pesar de las dos (2) oraciones anteriores, si Creative

Commons se ha identificado expresamente como el licenciador, tendrá todos los derechos y obligaciones del licenciador.

Salvo para el propósito limitado de indicar al público que la obra está licenciada bajo la CCPL, ninguna parte utilizará la marca registrada “Creative Commons” o cualquier marca registrada o insignia relacionada con “Creative Commons” sin su consentimiento por escrito. Cualquier uso permitido se hará de conformidad con las pautas vigentes en cada momento sobre el uso de la marca registrada por “Creative Commons”, en tanto que sean publicadas su sitio web (website) o sean proporcionadas a petición previa.

Puede contactar con Creative Commons en: <http://creativecommons.org/>.

Licencia GPL

The GNU General Public License Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular cir-

cumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the

sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

13. APENDICE III - CUENTAS Y DATOS DE ACCESO

En este Apéndice se detallan las diferentes cuentas que se han creado en los sistemas necesarios que para el correcto funcionamiento de la Aplicación.

SISTEMA	APP42	GMAIL	Parse
USER	pfcrafaelgonzalez@gmail.com	pfcrafaelgonzalez	pfcrafaelgonzalez@gmail.com
PASS	XXXXX	XXXXX	XXXXX

SISTEMA	FACEBOOK	TWITTER	Google AdMob
USER	pfcrafaelgonzalez@gmail.com	PFC Rafael González Gronberg.	pfcrafaelgonzalez@gmail.com
PASS	XXXXX	XXXXX	XXXXX
EMAIL	pfcrafaelgonzalez@gmail.com	pfcrafaelgonzalez@gmail.com	pfcrafaelgonzalez@gmail.com

SISTEMA	GOOGLE DEVELOPER	GOOGLE KEYSTORE
USER	CUENTA PERSONAL	Cuenta Personal Asociada a Keyhash de ordenador personal.
PASS		XXXXX
EMAIL		pfcrafaelgonzalez@gmail.com

* Por motivos de seguridad si se requiere la contraseña por favor contactar con el autor del proyecto en: 100059104@alumnos.uc3m.es

14. APENDICE IV – ICONOGRAFÍA E IMAGOTIPO:



Ilustración 127 - Logo Hispania Rectangular



Ilustración 128 Logo Hispania amarillo V.1



Ilustración 129 Logo Hispania amarillo V.2



Ilustración 126 Logo Hispania B/N



Ilustración 125 Logo Hispania Azul uc3m



Universidad
Carlos III de Madrid



Ilustración 130 Imagen Destacada para Google Play



Si has construido castillos en el aire, tu trabajo no se pierde; ahora coloca las bases debajo de ellos.

George Bernard Shaw
1856-1950. Escritor irlandés.

Ves cosas y dices, " Por qué?" Pero yo sueño cosas que nunca fueron y digo, "¿Por qué no?"

George Bernard Shaw
1856-1950. Escritor irlandés.

El progreso tecnológico sólo nos ha provisto de medios más eficientes para ir hacia atrás.

Aldous Huxley
1894-1963, Novelista y poeta inglés.

